

Universidad Carlos III de Madrid

Escuela Politécnica Superior



**APLICACIÓN WEB PARA LA GENERACIÓN DE CUESTIONARIOS
ONLINE Y GESTION DE RECURSOS E-LEARNING**

Proyecto Fin de Carrera

Ingeniería Técnica Informática de Gestión

Autor: **Alexandra González Vicente**
Tutora: **Isabel Segura Bedmar**
Fecha: **Febrero 2014**

Tabla de contenido

1. Introducción	4
1.1. Motivación	4
1.2. Objetivos	6
1.3. Estructura del Documento	7
2. Gestión del Proyecto	8
2.1. Gestión Software	8
2.1.1. Metodología de Desarrollo	8
2.1.2. Ciclo de Vida	9
2.2. Planificación Inicial	9
3. Análisis de Sistemas Similares	11
4. Análisis	13
4.1. Captura de Requisitos	13
4.2. Requisitos de Usuario	14
4.2.1. Funcionalidades del Sistema	14
4.2.2. Restricciones del Sistema	15
4.2.3. Usuarios del Sistema	15
4.2.4. Entorno Operacional	16
4.3. Requisitos del Software	16
4.3.1. Especificación de Requisitos	16
5. Diseño	26
5.1. Base de Datos	26
5.1.1. Esquema Conceptual	26
5.1.1.1. Diagrama E/R	26
5.1.1.2. Supuestos Semánticos	27
5.1.2. Esquema Lógico	29
5.2. Aplicación Software	29
5.2.1. Contexto	29
5.2.2. Arquitectura	30
5.2.3. Diseño de Componentes	32

6. Implementación y Pruebas	38
6.1. Base de datos	38
6.1.1. Selección de SGBD	38
6.1.2. Implementación y Pruebas	40
6.2. Aplicación Software	40
6.2.1. Selección del Lenguaje de Programación	40
6.2.2. Instalación del software inicial	41
6.2.3. Estructuración en Incrementos	41
6.2.4. Implementación de la Arquitectura	42
6.2.5. Implementación del Sistema	42
6.2.6. Pruebas	43
7. Planificación Final y Análisis de Costes	45
7.1. Planificación Final	45
7.2. Análisis de Costes	47
7.2.1. Recursos Humanos	47
7.2.2. Equipamiento	47
7.2.3. Fungibles	48
7.2.4. Otros	48
7.2.5. Coste Total	49
8. Conclusiones y Trabajo Futuro	50
8.1. Conclusiones	50
8.2. Trabajo Futuro	51
9. Glosario de Términos y Acrónimos	53
10. Referencias	56
Anexo 1 – Pruebas de Usabilidad	57
Anexo 2 – Manual de Instalación	61
Anexo 3 – Manual de Usuario	62
Perfil Alumno	63
Perfil Profesor	67
Perfil Administrador	70

1. Introducción

1.1. Motivación

Para competir en la economía post-industrial y globalizada del futuro, las empresas van a requerir que nuestras universidades formen profesionales con un perfil internacional, dotados de capacidad de aprendizaje autónomo, y familiarizados con el uso de las modernas tecnologías de la información y la comunicación.

En la sociedad del futuro los ciudadanos y compañías van a ser cada vez más exigentes con la calidad de los servicios que perciben y, para proporcionar estos servicios de alta calidad, serán necesarios mejores profesionales con competencias profesionales plenamente desarrolladas durante su educación superior. Las competencias y habilidades que el profesional necesita sólo se aprenden por medio de nuevos métodos didácticos que requieren la participación activa del estudiante.

Para lograr estas metas pueden usarse diversos métodos [1]:

- (1) Cada método tiene que animar procesos de reflexión, que incluyen tanto las experimentaciones como la incertidumbre en el espacio nuevo de enseñanza.
- (2) El método que ofrece un aprendizaje tanto teórico como práctico, además de incluir elementos de la vida fuera de las instituciones de educación.
- (3) El enfoque de aprendizaje basado en problemas, que consiste en plantear situaciones que podrían surgir en la vida real y buscar soluciones. Es adecuado para motivar a los estudiantes a integrar su conocimiento y generar sentido en lugar de acumular hechos y teoremas.
- (4) Para superar los dilemas que ocasionan los métodos avanzados de enseñanza y aprendizaje, se han desarrollado enfoques más estructurados bajo la marca «WELL» significando «Wechselseitiges Lehren und Lernen» (enseñanza y aprendizaje mutual). Una de las ideas fundamentales consiste en evitar problemas que puedan surgir en la colaboración entre estudiantes y ofrecer soluciones en caso de dificultades. Cuatro principios han guiado el desarrollo de los métodos WELL:
 - Ayudar a los estudiantes con estrategias adecuadas de aprendizaje, por ejemplo, debates, investigación, juego de roles o trabajo de campo.

- Estimular el que los estudiantes se enseñen los unos a los otros.
- Reconocer y retroalimentar los rendimientos, no de los miembros individuales, sino del conjunto social del equipo.
- Negociar reglas e iniciar procesos de reflexión.

El proceso de Bolonia tiene como objetivo la creación de un Espacio Europeo de Educación Superior (EEES), que facilitará la movilidad de los estudiantes, los preparará para su proyecto profesional y apoyará su desarrollo personal, así como la calidad de los estudios, respetando las garantías académicas y adaptándolas a las necesidades actuales de las empresas y de la sociedad actual, la cual requiere que nuestras universidades formen profesionales dotados de capacidad de aprendizaje autónomo, y familiarizados con el uso de las modernas tecnologías de la información y la comunicación [2].

El Proceso de Bolonia abandona la estructura tradicional de clases teóricas y examen final, para dar paso a un modelo centrado en el estudiante a través de su evolución en el aprendizaje y de la evaluación continuada por parte de los docentes, lo que comporta horas de trabajo personal y un peso importante de los trabajos en grupo. Esto representa una participación activa del estudiante que pasa de sujeto pasivo a actor de sus propios estudios. Esta metodología representa una adaptación, no sólo del estudiante sino también del profesor.

Aunque puedan realizarse exámenes finales, no serán los únicos métodos de evaluación existentes. Predominarán los procedimientos destinados a facilitar el seguimiento del aprendizaje del estudiante durante su vida universitaria (pruebas de respuesta breve, pruebas orales, trabajos, memorias, proyectos, pruebas de autoevaluación, escala de actitudes, técnicas de observación, etc.) [3]

El Plan Bolonia tiene algunas desventajas, y es que hacen falta más profesores para afrontar las nuevas formas de evaluación y los profesores deben invertir más tiempo para evaluar a los alumnos.

Por otro lado, los alumnos se ven obligados a afrontar una gran cantidad de trabajo de forma continuada, este hecho hace que sea complicado compatibilizar el estudio con el trabajo, algo que no ocurría anteriormente.

El nuevo perfil del profesorado universitario en el EEES

El perfil profesional del profesorado universitario en el nuevo contexto del EEES exige que los profesores afronten nuevas demandas y retos en formación y capacitación. No bastará con que sean expertos en su disciplina, deberán ser también especialistas en las metodologías necesarias para el aprendizaje de las mismas y esto exige formación y entrenamiento en nuevas metodologías didácticas orientadas al desarrollo de competencias en sus alumnos [4].

Surgen algunos inconvenientes ante la implantación del Plan Bolonia, el de mayor tamaño es la falta de medios en mitad de una crisis económica que reduce presupuestos, en un momento en el que el alumnado está volviendo a crecer y en un contexto que exige, en teoría, clases más pequeñas y más dedicación docente [5].

1.2. Objetivos

La finalidad de este proyecto es facilitar el seguimiento de la evaluación continua de los alumnos a través de una aplicación web para la asignatura “Estructura de Datos y Algoritmos”. Esta aplicación web permitirá a los alumnos formular sus dudas en forma de preguntas, y que estas sean respondidas por otros alumnos. Con esta aplicación se pretende proporcionar al profesorado una herramienta adicional para la evaluación de los alumnos, de una manera más ágil, eficiente y sencilla. Además, otros de los objetivos de la aplicación será motivar a los alumnos para que trabajen de manera continuada y de forma colaborativa en la asignatura y que resuelvan todas las dudas y problemas que se les planteen.

Además, se pretende fomentar el uso de las TIC en la docencia, ya que a pesar de los reconocimientos de las TIC para transformar los entornos educativos, su adopción ha sido menor o más lenta que en otros sectores de desarrollo de la sociedad. El uso de las TIC supone para los alumnos una forma de trabajo y aprendizaje más atractiva, que conlleva menos tiempo, acceso a múltiples recursos educativos, autoevaluación, mayor proximidad al profesor, compañerismo y colaboración. Para los profesores supone una fuente de recursos educativos para la docencia, orientación, investigación didáctica, facilidades para formar grupos de estudiantes, mayor relación con los alumnos, facilidad de evaluación y control, contacto con otros centros y profesores y liberación de trabajo monótono.

Resumen de la aplicación desarrollada:

Esta aplicación constará de tres perfiles: profesor, alumno y administrador. Cada uno de ellos podrá realizar tareas concretas y consultar distinta información que se detallará en próximos apartados.

La aplicación tendrá un menú con el temario de la asignatura donde se realizarán preguntas relacionadas con cada tema, que los alumnos deberán responder para recibir votaciones. Pueden realizar preguntas tanto los alumnos como el profesor de la asignatura y contestarán sólo los alumnos. La finalidad de que los alumnos realicen también preguntas sobre el temario es solventar las dudas que puedan surgirles. Cada respuesta puede recibir una o varias notas de alumnos y/o del profesor. Existen tres tipos de notas: nota dada por el profesor, nota dada por el alumno que realizó la pregunta (si aplica) y nota dada por el resto de alumnos a una cierta respuesta. Al final de curso, el profesor puede calcular de manera automática la nota final de los alumnos a partir de sus notas medias.

Tecnologías aprendidas para el desarrollo de la aplicación:

Uno de los objetivos, además del uso de la aplicación desarrollada, ha sido aprender nuevas tecnologías como JSP, nuevas herramientas, Apache Tomcat, el gestor de bases de datos MySQL y un nuevo entorno de desarrollo, Netbeans. Todo lo anteriormente mencionado se ha debido aprender, sin ninguna base previa, para enfrentarse a este proyecto.

1.3. Estructura del Documento

El documento actual se estructura en nueve secciones principales y tres anexos.

La primera sección detalla la motivación y los objetivos de este proyecto. La segunda describe la gestión de proyecto realizada. La tercera, habla sobre el estado del arte. Las secciones cuarta, quinta y sexta describen las fases de análisis, diseño e implementación respectivamente. La sección séptima proporciona la planificación final del proyecto y el análisis de costes realizado. La octava, ofrece conclusiones e ideas para trabajos futuros. La sección nueve comprende el glosario de término y la diez la bibliografía.

Por último, se presentan tres anexos: manual de instalación de la aplicación, pruebas realizadas a usuarios del sistema y un manual de usuario.

2. Gestión del Proyecto

La buena gestión de un proyecto es la base para lograr su éxito. En primer lugar se deben analizar las fases que guiarán el proyecto, las tareas que se llevarán a cabo en cada una de las fases y la documentación que hay que generar. En resumen, es necesario desarrollar una planificación inicial completa del ciclo de vida del proyecto.

2.1. Gestión Software

Para gestionar este proyecto se ha utilizado una metodología de desarrollo de software, planificando adecuadamente los procesos y tareas a realizar a lo largo del ciclo de vida del proyecto.

2.1.1. Metodología de Desarrollo

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Seleccionar la metodología apropiada para cada proyecto es una tarea básica para garantizar su éxito. Las metodologías de desarrollo de software se pueden clasificar en dos grandes grupos:

- **Pesadas o tradicionales:** Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

Esta metodología se utiliza principalmente cuando los equipos de desarrollo son grandes (más de 10 integrantes), los requisitos están bien definidos y la duración del proyecto es superior a un año.

- **Ligeras o ágiles:** Esta metodología se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados. Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación y resulta más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan.

Esta metodología se utiliza principalmente cuando los equipos de desarrollo son pequeños (menos de 10 integrantes), los requisitos no están bien definidos (son cambiantes) y la duración del proyecto es inferior a un año.

Puesto que el proyecto que se va a desarrollar tiene una duración menor de un año, con un equipo de un integrante y con requisitos cambiables durante su desarrollo, la metodología a seguir sería una metodología ligera o ágil. Sin embargo, las metodologías ligeras no proporcionan guías ni técnicas para generar documentación de calidad, algo que se desea para este proyecto. Por este motivo, se ha decidido seguir una metodología híbrida (decisión bastante habitual en proyectos con equipos de desarrollo de menos de 3 integrantes), con lo mejor de las metodologías tradicionales y lo mejor de las metodologías ligeras [6]:

La metodología tradicional seguida será RUP, se trata de una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

La metodología ligera seguida es la que más destaca de los procesos ágiles en ingeniería de software, XP (Extreme Programming). Se caracteriza por un desarrollo interactivo e incremental, realización de pruebas unitarias continuas, simplicidad en el código, corrección de errores antes de añadir nuevas funcionalidades, integración del equipo de programación con el cliente y refactorización del código.

2.1.2. Ciclo de Vida

El desarrollo de software va unido a un ciclo de vida compuesto por una serie de etapas que comprenden todas las actividades, desde el momento en que surge la idea de crear un nuevo producto software, hasta aquel en que el producto deja definitivamente de ser utilizado por el último de sus usuarios.

2.2. Planificación Inicial

La planificación inicial es uno de los puntos más fuertes de un proyecto, ya que servirá de guía para su desarrollo. Cuanto menor sea el número de cambios que sufra esta

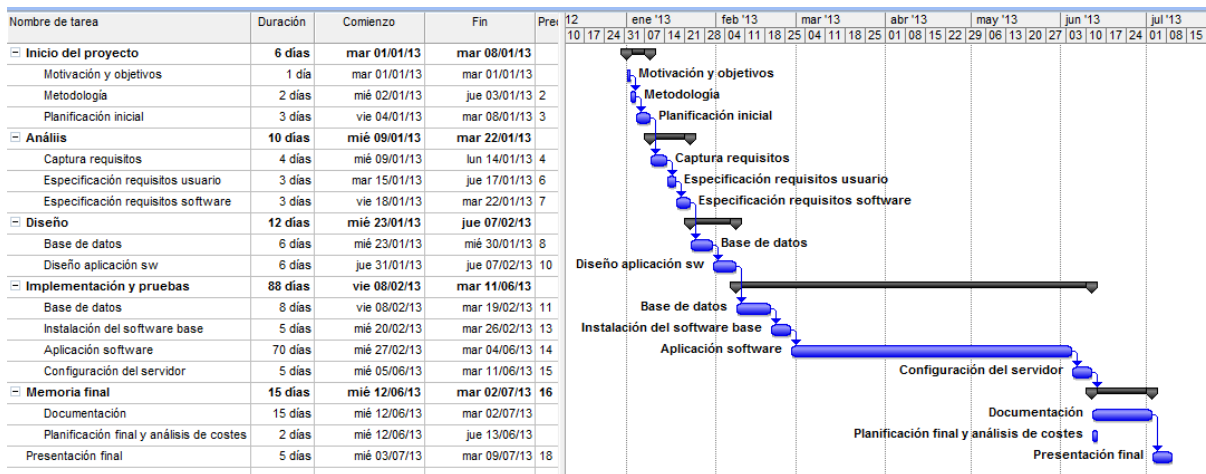
planificación a lo largo del desarrollo del proyecto, menor será el coste de éste y mayor será el beneficio.

Por otro lado, la planificación ayuda a estimar el coste del proyecto. Al no tratarse de un proyecto comercial, este coste se detallará al finalizar el proyecto en el apartado relacionado con el análisis de costes.

Para realizar la planificación del proyecto se han tenido en cuenta los hechos siguientes:

- Se trabaja de lunes a viernes, 4 horas cada día.
- El equipo de desarrollo está formado por un único integrante sin experiencia en este tipo de proyectos. Según las necesidades, este integrante tomará uno de los siguientes roles: gestor de proyecto, analista/diseñador, diseñador de interfaces y programador.

La planificación inicial del proyecto abarca desde el 1 de enero hasta el 30 de junio de 2013 y las tareas en las que se divide coinciden con las definidas en este documento. La planificación final se detallará en el apartado 7.1. A continuación se muestra el diagrama de Gantt generado inicialmente:



3. Análisis de Sistemas Similares

En esta sección se ofrece una visión global de las tecnologías y recursos que pueden ser relacionados con este proyecto.

En primer lugar, uno de los productos altamente relacionado con el proyecto son los foros. Los foros dan soporte a discusiones u opiniones en línea, permitiendo al usuario poder expresar su idea o comentario respecto al tema tratado. Los foros virtuales son un aspecto muy interesante en el proceso de enseñanza-aprendizaje universitario, dando gran importancia a las actividades colaborativas y sirviendo de apoyo a las clases presenciales. Un foro puede usarse como forma de diálogo entre usuarios, pero las respuestas no pueden votarse, y por consiguiente no se puede obtener una calificación media de los usuarios. Un ejemplo de foro relacionado con asignaturas universitarias de la Universidad Carlos III de Madrid es el del Laboratorio del Departamento de Informática⁷.

En este proyecto es primordial el intercambio de ideas entre alumnos y profesor, y aunque funcione como un foro, existen numerosos detalles que lo diferencian, entre ellos que las respuestas pueden ser votadas y cuando el profesor lo desee puede calcular las notas finales en función de las votaciones que crea convenientes y enviarlas por email.

Otro tipo de recursos altamente relacionados con el proyecto son los blogs, un sitio web que se actualiza periódicamente y que recopila cronológicamente textos de varios autores. En un blog los lectores pueden escribir sus comentarios y el autor darles respuesta, de forma que es posible establecer un diálogo. En algunos blogs se pueden votar los comentarios de los lectores, pero normalmente el voto puede ser “positivo” o “negativo”, no se lleva un recuento de votos por usuario. En Internet existen numerosos blogs relacionados con todo tipo de temas, en la Universidad Carlos III de Madrid por ejemplo hay un blog de Delegación de Estudiantes [8].

Dos recursos muy relacionados con preguntas y respuestas con votaciones son Yahoo Respuestas [9] y OSQA [10].

Yahoo Respuestas es un sistema en el que los usuarios proponen preguntas (dudas, problemas, opiniones), y los lectores responden, de forma que el usuario que realizó la pregunta puede votar sus respuestas con una puntuación de 1 a 5. Además, existe un recuento de votos por usuario. En este caso es un producto destinado a todo tipo de temas, donde no se ofrece la posibilidad de controlar los usuarios y las votaciones válidas como ocurre en el proyecto que se ha desarrollado, ni de calcular una nota media en

función de las votaciones de varios tipos de usuarios. Es importante destacar que la aplicación desarrollada en este proyecto, permite al profesor calcular la nota final de los alumnos a partir de ponderaciones de las notas medias que considere oportunas.

OSQA es un sistema en el que los usuarios pueden formular y contestar preguntas, comentar y votar las preguntas de los demás y sus respuestas. Estos sistemas no están enfocados a estudiantes, ni para un tema en concreto como ocurre en la aplicación implementada en este proyecto; además, todos los usuarios tienen el mismo perfil, y lo más importante, como en el caso anterior, OSQA no permite calcular una nota final a partir de ponderaciones de las notas medias.

Otros recursos relacionados que también están dirigidos a la educación pueden ser Moodle [11] y eFront [12]. Moodle es un sistema que ayuda a los educadores a crear comunidades virtuales de aprendizaje, dirigido a cualquier nivel de estudios: cursos, escuelas de educación infantil y primaria, etc. eFront es una plataforma de aprendizaje que cuenta con tres perfiles de usuarios (profesor, alumno y administrador), al igual que la aplicación desarrollada en este proyecto, pero que está enfocada simplemente a organizar material pedagógico, publicar pruebas y exámenes, comunicarse con los alumnos, etc., no a realizar un seguimiento de los alumnos y evaluarlos a través de la propia aplicación.

Además, comentar el uso de Web Quest [13], una herramienta que fomenta la realización de tareas didácticas mediante una investigación guiada, principalmente con recursos de Internet. Las tareas que se plantean deben consistir en algo más que en contestar a simples preguntas o reproducir lo que hay en la pantalla, las Web Quest tratan de incitar al pensamiento creativo para solucionar problemas, aumentar la capacidad de análisis y síntesis. La aplicación desarrollada es algo más simple y se utiliza para realizar preguntas con el fin de evaluar a los alumnos, ayudarles a solventar dudas y animarles a participar activamente en la asignatura, intentando además que estén al día de los temas impartidos en la materia.

Al no existir ninguna aplicación tan completa y enfocada a la evaluación de usuarios como la que se desarrolla en este proyecto, parece una idea interesante llevarla a cabo y cubrir su falta, ofreciendo así un servicio de ayuda a estudiantes y profesorado.

4. Análisis

La primera fase del ciclo de vida de este proyecto es el análisis. Esta fase consiste en el estudio de las necesidades de los usuarios para llegar a una definición del sistema. El resultado obtenido es un documento de especificaciones de requisitos que describe lo que el sistema debe hacer.

4.1. Captura de Requisitos

La captura de requisitos es la actividad mediante la que se extraen las necesidades que debe cubrir el sistema, utilizando para ello diferentes técnicas:

- Entrevistas: El objetivo de las entrevistas es que el cliente narre su problemática y guíe al analista para determinar requisitos.
- Cuestionarios: El analista genera un cuestionario de preguntas que el cliente debe contestar.
- Discusiones: Estas entrevistas pretenden que el analista y cliente mantengan una discusión sobre la problemática existente y obtener así requisitos del sistema.
- Introspección: El analista se pone en el lugar del usuario final del sistema y crea una lista con las principales funcionalidades que debe cumplir. Una vez finalizada esta lista se realiza una discusión con el cliente.
- Casos de uso: Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario para conseguir un objetivo específico.

Durante la realización del proyecto se han obtenido varias versiones de la especificación de requisitos de usuario. Las primeras técnicas empleadas para capturar nuevos requisitos fueron una serie de entrevistas de final abierto y cuestionarios. Las discusiones siguientes se utilizaron para revisar las descripciones de los requisitos ya capturados y asegurar que su especificación era clara y sin ambigüedades.

Una vez realizada la captura inicial de requisitos de usuario, se realizó una sesión de introspección. Como consecuencia se obtuvieron nuevos requisitos (RF-04, RF-09 y RF-16) y se ampliaron otros (RF-08, RF-10, RF-12).

Aprobada la especificación de requisitos de usuario, se continuó con la captura de requisitos del software. La principal técnica utilizada es la realización de diagramas de

casos de uso. Además, se estudiaron las restricciones operativas, de entorno e implementación.

Para finalizar la fase de análisis tuvieron lugar una serie de discusiones con el fin de completar la especificación de requisitos del software y perfeccionar las descripciones de los requisitos capturados. Esta especificación será la base para la fase de diseño.

4.2. Requisitos de Usuario

Los requisitos de usuario proporcionan una especificación de alto nivel de las funcionalidades y restricciones que el cliente y/o usuario final necesita en el sistema resultante. Describen los requerimientos funcionales y no funcionales de tal forma que sean comprensibles por los usuarios del sistema que no posean un conocimiento técnico detallado.

4.2.1. Funcionalidades del Sistema

Existen tres perfiles de usuario: alumno, profesor y administrador. Cada uno de estos perfiles puede realizar una serie de tareas. Las funcionalidades del sistema resultante son las siguientes:

- Registrarse como usuario.

➤ Perfil Alumno

- Formular preguntas por categoría.
- Proponer respuestas a las preguntas.
- Votar respuestas de otros alumnos.
- Consultar las 5 preguntas que se han respondido más recientemente.
- Consultar notas recibidas.
- Modificar contraseña.

➤ Perfil Profesor

- Formular preguntas por categoría.
- Comentar preguntas y respuestas.
- Votar respuestas.

- Consultar las 5 preguntas que se han respondido más recientemente.
- Listar datos de alumnos.
- Validar alumnos registrados.
- Solicitar borrado de alumnos.
- Denunciar preguntas y respuestas.
- Calcular notas finales en función de distintos criterios.
- Enviar emails con la nota final de cada alumno.

➤ Perfil Administrador

- Borrar alumnos.
- Borrar preguntas.
- Borrar respuestas.

4.2.2. Restricciones del Sistema

Las restricciones que el sistema debe cumplir se muestran a continuación:

- Debe tener interfaz gráfica de usuario.
- No debe requerir la instalación de aplicaciones o componentes.
- Debe visualizarse y funcionar correctamente en Internet Explorer y Mozilla Firefox.
- Debe cumplir las disposiciones recogidas en la Ley Orgánica de Protección de Datos de Carácter Personal.
- El impacto de futuras ampliaciones debe ser mínimo.

4.2.3. Usuarios del Sistema

El sistema puede ser visualizado únicamente por todos los usuarios que tengan acceso a la aplicación, es decir, que se hayan registrado, hayan sido validados por el profesor de la asignatura y hayan recibido un email de confirmación.

El nivel de conocimientos informáticos requerido para utilizar correctamente la aplicación es medio.

4.2.4. Entorno Operacional

El entorno operacional de la aplicación es simple: un ordenador con un entorno de ejecución adecuado, recursos suficientes y con acceso a Internet.

4.3. Requisitos del Software

Los requisitos del software establecen con precisión los servicios y restricciones que el sistema debe cumplir. Son descripciones más detalladas que los requisitos de usuario. Sirven como base para definir el contrato de la especificación del sistema y, por lo tanto, debe ser una especificación completa y consistente del sistema. Son utilizados por los ingenieros de software como punto de partida para el diseño del sistema.

4.3.1. Especificación de Requisitos

Con el fin de proporcionar una especificación clara, cada requisito se detalla en una tabla con el siguiente formato:

ID		Tipo	
Nombre			
Descripción			
Relaciones		Dependencias	
Verificable		Criticidad	
Necesidad		Conveniencia	

Tabla 1 - Formato de Requisito del Software

Los campos de la tabla son:

- **ID:** Identificador del requisito. Los requisitos funcionales y no funcionales se especifican como RF y RNF, respectivamente, seguido de un número que coincide con la posición en la que aparecen en la especificación de los siguientes apartados.
- **Tipo:** Tipo de requisito. Los valores que puede tomar son: funcional, interfaz, portabilidad y escalabilidad.
- **Nombre:** Nombre corto y descriptivo que permite al lector hacerse una idea de la finalidad del requisito.

- **Descripción:** Descripción detallada del requisito.
- **Relaciones:** Requisitos de software con los que se relaciona este requisito. Si existen relaciones, se especificarán los identificadores de los requisitos con los que se relaciona. Si no existen relaciones, se escribirá “-”.
- **Dependencias:** Requisitos de software de los que depende este requisito. Si existen dependencias, se especificarán los identificadores de los requisitos de los que depende. Si no existen dependencias, se escribirá “-”.
- **Verificable:** Indica si el requisito puede ser verificado o no. Los únicos valores posibles son “Sí” y “No”.
- **Criticidad:** Indica cuán crítico es para el cliente el incumplimiento del requisito. Los únicos valores posibles son “Alta”, “Media” y “Baja”.
- **Necesidad:** Indica cuán necesario es para el cliente el cumplimiento del requisito. Los únicos valores posibles son “Esencial”, “Conveniente” y “Opcional”.
- **Conveniencia:** Indica cuán conveniente es para el cliente el cumplimiento del requisito. Los únicos valores posibles son “Alta”, “Media” y “Baja”. En el caso de particular de requisitos esenciales, la conveniencia se establece siempre como alta.

4.3.1.1. Requisitos Funcionales

ID	RF - 01	Tipo	Funcional
Nombre	Crear usuarios		
Descripción	<p>Los alumnos pueden registrarse en la aplicación, pero serán validados por el profesor antes de poder acceder a la misma. Deberán introducir el nombre, apellidos, e-mail, NIA, contraseña y DNI.</p> <p>Existe un único profesor.</p>		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 02	Tipo	Funcional
Nombre	Borrar usuarios		
Descripción	<p>El profesor puede solicitar el borrado de usuarios al administrador. Será desde el perfil administrador donde se den de baja.</p> <p>Las bajas serán lógicas para no perder información de notas de otros alumnos.</p>		
Relaciones	-	Dependencias	RF - 01
Verificable	Sí	Criticidad	Alta
Necesidad	Conveniente	Conveniencia	Alta

ID	RF - 03	Tipo	Funcional
Nombre	Validar usuarios		
Descripción	El profesor debe validar a los usuarios que se registren para que puedan acceder al sistema.		
Relaciones	-	Dependencias	RF - 01
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 04	Tipo	Funcional
Nombre	Enviar correo validación		
Descripción	El sistema enviará automáticamente a los alumnos un correo una vez hayan sido validados por el profesor.		
Relaciones	-	Dependencias	RF-01, RF-03
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Alta

ID	RF - 05	Tipo	Funcional
Nombre	Cambiar contraseña		
Descripción	Los alumnos tienen la opción de modificar su contraseña de acceso al sistema.		
Relaciones	-	Dependencias	RF-01, RF-03
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Media

ID	RF - 06	Tipo	Funcional
Nombre	Listar alumnos		
Descripción	<p>El sistema mostrará un listado de alumnos con los siguientes datos: nombre y apellidos, email, NIA y estado (validado o no validado).</p> <p>Este listado puede consultarse sólo desde el perfil profesor.</p>		
Relaciones	RF-01	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 07	Tipo	Funcional
Nombre	Formular preguntas		
Descripción	Tanto el profesor como los alumnos podrán formular preguntas abiertas sobre la materia. Estas preguntas pertenecen a una categoría (tema de la asignatura).		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 08	Tipo	Funcional
Nombre	Proponer respuestas		
Descripción	<p>Los alumnos podrán responder a las preguntas formuladas tantas veces como consideren oportuno.</p> <p>Los profesores no pueden responder, pero pueden comentar las preguntas y respuestas.</p>		
Relaciones	-	Dependencias	RF-07
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 09	Tipo	Funcional
Nombre	Denunciar preguntas y respuestas.		
Descripción	<p>El profesor puede denunciar las preguntas y respuestas que considere para que el administrador las elimine.</p> <p>Las bajas serán lógicas para no perder información de notas de los alumnos.</p>		
Relaciones	RF-01	Dependencias	RF-07, RF-08
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Media

ID	RF -10	Tipo	Funcional
Nombre	Calificar respuestas		
Descripción	<p>El profesor y los alumnos podrán calificar la calidad de la respuesta con una puntuación de 0 a 5.</p> <p>Cada respuesta tiene asociadas varias notas: nota del profesor, nota del alumno que propone la pregunta y las notas dadas por el resto de alumnos según lo útil que haya sido para ellos dicha respuesta.</p> <p>Si la pregunta es planteada por el profesor no existirá nota del alumno.</p>		

	Un alumno no puede calificar una respuesta dada por él mismo ni puede votar una respuesta más de una vez.		
Relaciones	-	Dependencias	RF-08
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 11	Tipo	Funcional
Nombre	Ver notas recibidas		
Descripción	El alumno puede consultar las notas que ha recibido en cada respuesta. Se muestran la nota del profesor, nota del alumno que propuso la pregunta y la media de las nota recibidas por el resto de alumnos.		
Relaciones	-	Dependencias	RF-03, RF-08, RF-10
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 12	Tipo	Funcional
Nombre	Calcular nota media calificaciones		
Descripción	<p>El profesor puede consultas las notas de los alumnos y calcular las notas finales a partir de la media de las notas recibidas en sus respuestas.</p> <p>Se proporcionará una lista con las opciones que desea que se tengan en cuenta para calcular la nota media final y las ponderaciones de cada una de las opciones.</p> <p>Las opciones a elegir serán las notas del profesor, del alumno que formuló las preguntas o del resto de alumnos. El profesor podrá seleccionar una, dos o las tres opciones, y ponderarlas.</p>		
Relaciones	RF-10	Dependencias	RF-08, RF-10
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 13	Tipo	Funcional
Nombre	Enviar nota final		
Descripción	Cuando el profesor calcula la nota final de los alumnos, puede enviar un email automáticamente a cada uno de ellos para informarles de la nota recibida.		
Relaciones	RF-11	Dependencias	RF-12
Verificable	Sí	Criticidad	Alta
Necesidad	Conveniente	Conveniencia	Alta

ID	RF - 14	Tipo	Funcional
Nombre	Acceso		
Descripción	Los usuarios registrados y validados deben acceder mediante su NIU y usando la contraseña recibida en el correo de validación.		
Relaciones	RF-01, RF-03	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 15	Tipo	Funcional
Nombre	Visualizar preguntas y respuestas por temas		
Descripción	<p>Cada pregunta y respuesta pertenecerá a una categoría y puede ser visualizado por los alumnos y el profesor.</p> <p>En cada categoría se mostrarán las preguntas que pertenezcan a esta y al pulsar en una pregunta concreta aparecerán las respuestas recibidas.</p>		
Relaciones	-	Dependencias	RF-07, RF-08
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 16	Tipo	Funcional
Nombre	Pantalla inicial		
Descripción	<p>La pantalla principal que aparece al iniciar sesión debe mostrar las 5 preguntas que hayan obtenido respuestas más recientemente.</p> <p>Esta pantalla aparecerá al acceder con perfil alumno y profesor.</p>		
Relaciones	-	Dependencias	RF-08
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RF - 17	Tipo	Funcional
Nombre	Sesiones abiertas		
Descripción	Un usuario sólo podrá tener una sesión abierta en la misma máquina.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Conveniente	Conveniencia	Alta

4.3.1.2. Requisitos No-Funcionales

ID	RNF - 01	Tipo	Interfaz
Nombre	Interfaz gráfica		
Descripción	La aplicación debe ofrecer una interfaz gráfica de fácil comprensión y aprendizaje, cumpliendo los principios de usabilidad.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RNF - 02	Tipo	Portabilidad
Nombre	Multiplataforma		
Descripción	La aplicación debe poder implantarse y ejecutarse tanto en Windows como en Linux.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RNF - 03	Tipo	Escalabilidad
Nombre	Visibilidad navegador		
Descripción	El sistema debe visualizarse y funcionar correctamente en Internet Explorer y Mozilla Firefox.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RNF - 04	Tipo	Externo
Nombre	Privacidad		
Descripción	El sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Protección de Datos de Carácter Personal.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

ID	RNF - 05	Tipo	Escalabilidad
Nombre	Futuras ampliaciones		
Descripción	El impacto de futuras ampliaciones debe ser mínimo. El sistema se desarrollará en módulos independientes.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Conveniente	Conveniencia	Alta

5. Diseño

Una vez conocidos exactamente los requisitos, se lleva a cabo la fase de diseño, que debe proporcionar una completa idea del sistema, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación. Esta fase consta, en este caso, de dos grandes tareas: diseño de la base de datos y diseño de la aplicación software.

5.1. Base de Datos

La base de datos se ha elegido de acuerdo a su modelo de administración de datos. El uso de una base de datos relacional resulta el más adecuado para modelar problemas reales y administrar datos dinámicamente. Una base de datos relacional cumple con el modelo de datos relacional, basado en la lógica de predicados y en la teoría de conjuntos. La idea fundamental es el uso de relaciones, que podrían considerarse una forma de organizar conjuntos de datos.

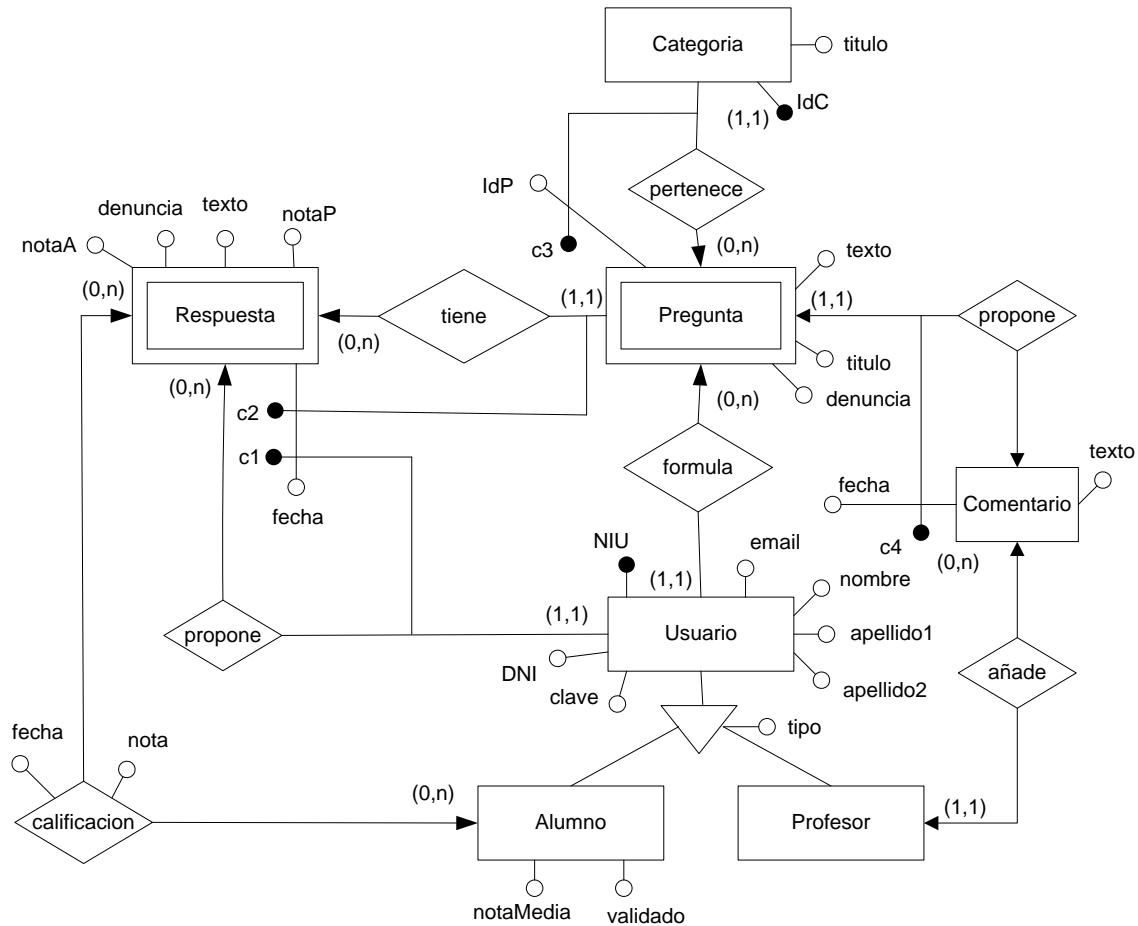
En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. Además, existen numerosos gestores gratuitos disponibles en el mercado.

5.1.1. Esquema Conceptual

Un esquema conceptual representa la lógica de la base de datos, es una descripción de alto nivel de su estructura, los datos que la componen y sus relaciones.

5.1.1.1. Diagrama E/R

El diagrama de entidad-relación representa entidades que intervienen en el sistema de información, así como sus interrelaciones, atributos y propiedades [14].



5.1.1.2. Supuestos Semánticos

Los supuestos semánticos asociados al esquema conceptual de la base de datos de este proyecto son los siguientes:

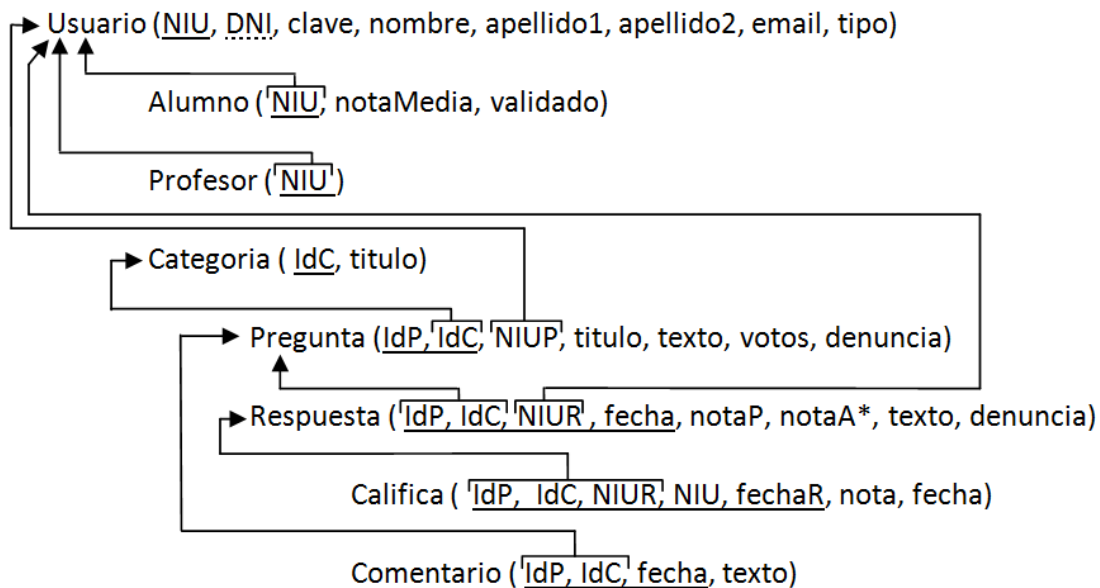
1. El atributo *titulo* de la entidad *Categoría* debe tomar alguno de los siguientes valores: “*Concepts of data structure and algorithms*”, “*Analysis os algorithms*”, “*Recursivity*”, “*Lists, stacks and queues*”, “*Trees*” y “*Java*”.
2. Los atributos *IdP* de *Pregunta* e *IdC* de *Categoría* son autoincrementales.
3. El atributo *tipo* de la entidad *Usuario* debe tomar valor “*P*” o “*A*”.
4. El atributo *validado* de la entidad *Alumno* debe tomar alguno de los siguientes valores: “*si*” o “*no*”, que podrá modificar el usuario tipo P.

5. Los atributos *notaP* y *notaA* de la entidad *Respuesta* deben tomar valores enteros de 0 a 5. Estos atributos hacen referencia a la nota recibida por el profesor y por el alumno que realizó la pregunta respectivamente. Pueden ser ambos nulos.
6. El atributo *notaA* de la entidad *Respuesta* tendrá valor nulo si una pregunta ha sido realizada por el profesor.
7. Una calificación consta de dos NIU, el del alumno que propuso la respuesta y el alumno que la califica.
8. El atributo *nota* de la entidad *Respuesta* debe tomar valor de 0 a 5. Esta nota hace referencia a la nota recibida por un alumno distinto al que respondió. Se utiliza para valorar la utilidad de esa respuesta para el resto de estudiantes.
9. El atributo *notaMedia* de la entidad *Alumno* debe tomar un valor decimal de 0 a 5. Su valor será nulo hasta que se calcule la nota media final de cada alumno.
10. El atributo *NIU* de la entidad *Usuario* puede usarse como clave alternativa.
11. El atributo *denuncia* de las entidades *Respuesta* y *Pregunta* puede tomar los valores “N” o “S”.
12. Un alumno no puede calificarse a sí mismo, por lo que el NIU de la entidad *Respuesta* debe ser diferente al NIU de la entidad *Alumno* para una calificación.
13. Un alumno podrá responder varias veces a una pregunta.
14. Un alumno sólo podrá calificar una vez cada respuesta.
15. Un alumno no puede responder a una pregunta propuesta por él mismo.
16. El atributo *notaMedia* de la entidad *Alumno* se calcula como la media de las notas obtenidas en las respuestas dadas por dicho alumno.
17. El atributo *NIU* de la entidad *Usuario* debe tomar valores de 100000000 a 100099999 cuando el tipo es “A” (es alumno) y de 400000000 a 400099999 para tipo “P” (es profesor).
18. El atributo *validado* de la entidad *Alumno* tendrá valor “no” por defecto hasta que el profesor realice su validación.
19. El atributo *fecha* de la entidad *Respuesta* debe ser inferior a la fecha actual.
20. El atributo DNI de la entidad *Usuario* es una clave alternativa.

5.1.2. Esquema Lógico

Un esquema lógico es una descripción de la estructura de la base de datos que puede procesar un SGBD. Así como el esquema conceptual es independiente del SGBD, el esquema lógico depende del tipo elegido (relacional, de redes, jerárquico...), pero no de un SGBD específico.

5.1.2.1. Grafo Relacional



5.2. Aplicación Software

La importancia del diseño del software se puede definir en una sola palabra, calidad, dentro del diseño es donde se fomenta la calidad del proyecto. El diseño es la única manera de materializar con precisión los requerimientos del cliente.

5.2.1. Contexto

El contexto de una aplicación hace referencia a los sistemas con los que se relaciona, en este caso no interactúa con ningún sistema externo.

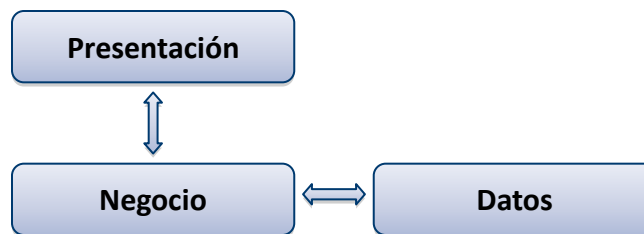
5.2.2. Arquitectura

La elección de la arquitectura en la que se va a basar un sistema es uno de los pasos iniciales, indica las unidades en las que se organizarán los componentes y las relaciones entre ellos.

Destacan las arquitecturas por capas, cuya finalidad reside en la división de la lógica de negocio de la presentación. La principal ventaja de estas arquitecturas es que, al modularizar, en caso de modificación, se puede atacar sólo uno de los niveles. En la actualidad, las arquitecturas más utilizadas son las que se detallan a continuación [15]:

Arquitectura en 1 capa: todos los componentes se encuentran al mismo nivel, de tal forma que la lógica de presentación y la de negocio no se separan. Esta arquitectura es muy sencilla de implementar pero también muy difícil de mantener y de ampliar, debido a la alta dependencia entre componentes y a la falta de estructuración del sistema.

Arquitectura en 3 capas: el sistema se divide en tres capas dispuestas en una pila como la de la ilustración siguiente:



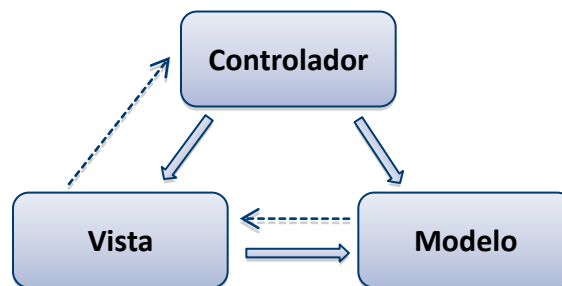
- **Presentación:** es la capa que presenta el sistema al usuario (interfaz gráfica), le comunica la información y captura la que proporcione el usuario.
- **Negocio:** contiene la información de los programas que se ejecutan y es donde se establecen todas las reglas que deben cumplirse.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio

La capa de presentación recibe eventos del usuario y se los pasa a la capa de negocio. Como consecuencia, la capa de negocio ejecuta alguna operación. Si la operación requiere

manejar datos, la capa de negocio establece comunicaciones con la capa de datos. Cuando la capa de negocio decida que el usuario debe ser informado de algo, pasará la información correspondiente a la capa de presentación, que se encargará de mostrar dicha información de la manera más adecuada.

Implementar esta arquitectura no es una tarea compleja, pero sí conlleva un esfuerzo importante. Sin embargo, un sistema con esta arquitectura puede ser mantenido y ampliado con un esfuerzo mínimo.

Arquitectura MVC: el sistema se divide en tres capas dispuestas en un triángulo como el de la ilustración siguiente:



Esta arquitectura es parecida a la arquitectura en 3 capas excepto por las propiedades siguientes:

- El modelo es la unión de la capa de negocio y de la capa de datos presentes en la arquitectura en 3 capas.
- La vista y el modelo se pueden comunicar entre sí: el modelo puede actualizar la visión que la vista tiene de él y la vista puede preguntar al modelo su estado actual.
- El controlador transforma los eventos del usuario generados en la vista en llamadas al modelo.

De estas tres arquitecturas, se ha decidido implementar la arquitectura en 3 capas, haciendo que el sistema tenga un mantenimiento y una complejidad de ampliación mínima. La forma de implementarlo consiste en utilizar HTML para la parte de presentación y Java para la parte de negocio y relación con la base de datos.

5.2.3. Diseño de Componentes

Después de elegir la arquitectura que implementará el sistema, se debe realizar el diseño de la aplicación.

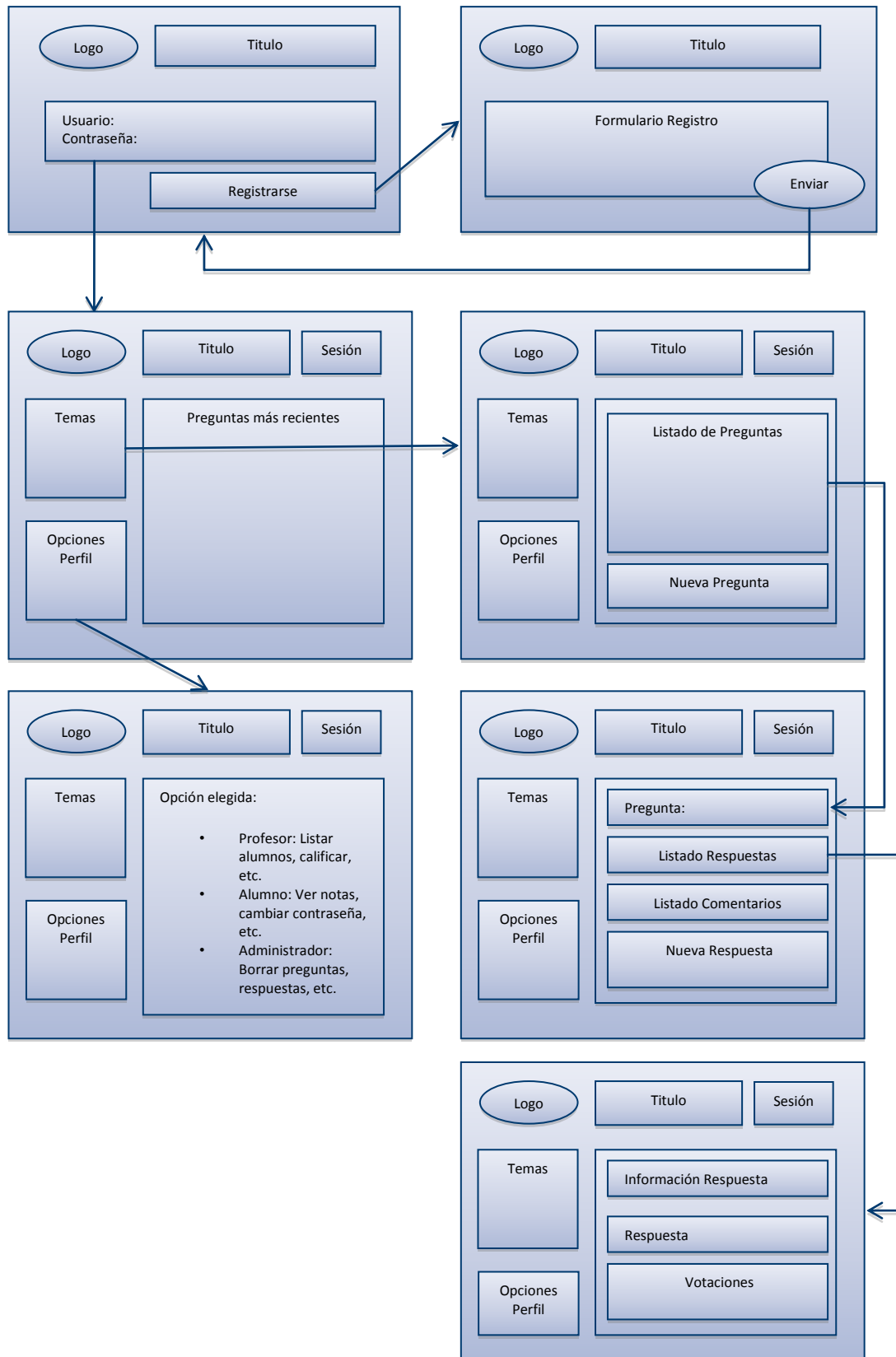
Existen multitud de modelos utilizados para el diseño de aplicaciones: diagramas de actividades, de clases, de objetos, de interacción, de componentes, etc. Según las necesidades y circunstancias de un proyecto, se usarán unos u otros de los modelos mencionados.

Este proyecto se ha decidido modelar mediante un diagrama de componentes, teniendo en cuenta la metodología seguida y los recursos disponibles. Este diagrama de componentes se complementa con una especificación de responsabilidades.

5.2.3.1. Estructura del prototipo

Antes de comenzar con el diseño de la funcionalidad del sistema, se elaboró un wireframe o prototipo visual del diseño de la interfaz, en el que puede contemplarse la arquitectura y las relaciones entre las páginas de la aplicación. La elaboración de este prototipo simplifica la complejidad del diseño y ahorra tiempo en su desarrollo.

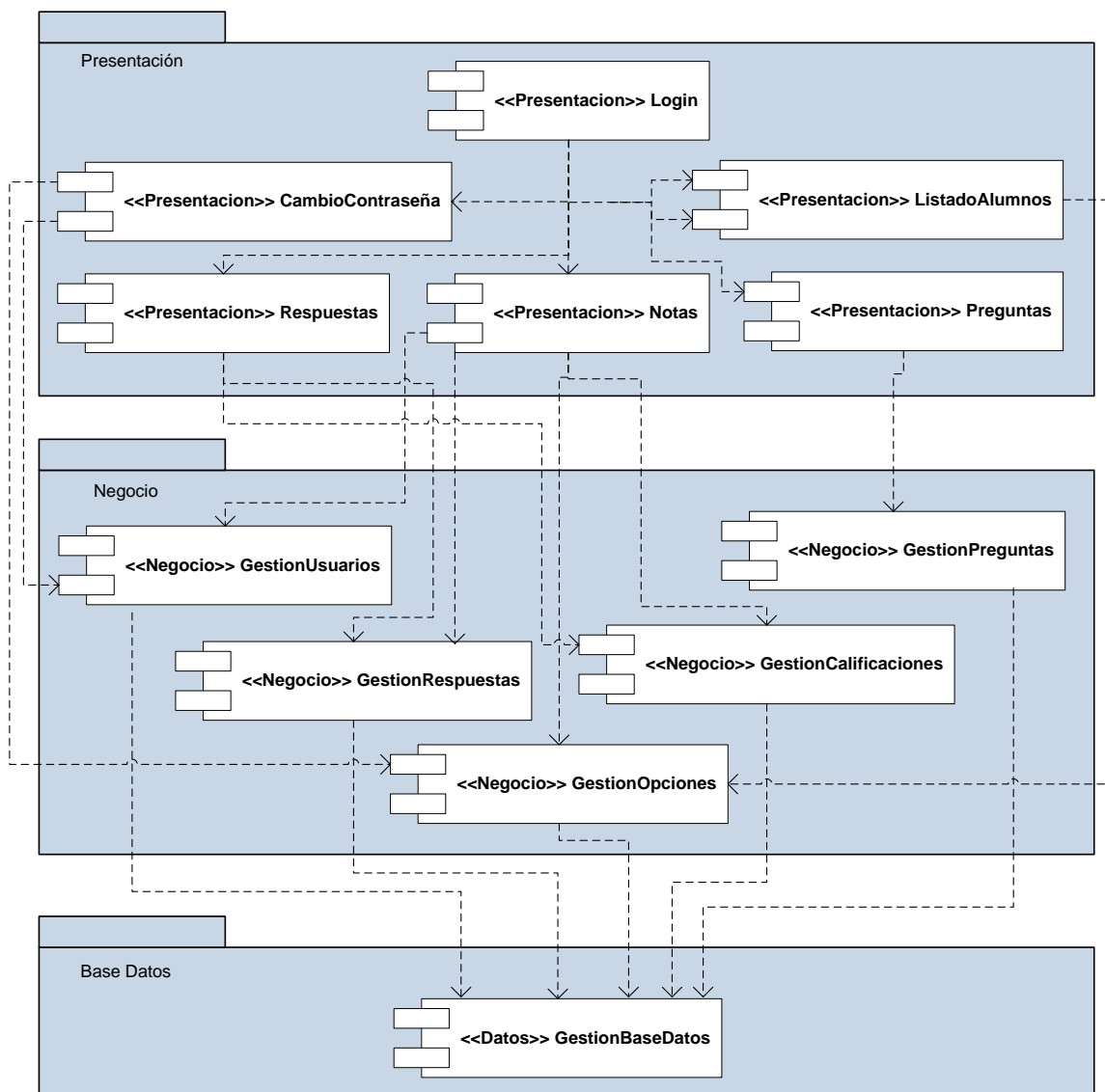
En esta fase se mostró el prototipo a varios usuarios con el objetivo de conseguir un diseño que fuera lo más usable posible y, sobre todo, cuya navegación resultara lógica e intuitiva. En la fase de pruebas, tras realizar la implementación, es donde se consigue proporcionar al desarrollador unos resultados definitivos, pero estas pruebas de usabilidad sobre el prototipo resultan muy útiles para realizar un mejor diseño de la aplicación.



5.2.3.2. Diagrama de Componentes

Un diagrama de componentes es un diagrama tipo UML que representa la división de un sistema de software en componentes y muestra las dependencias entre ellos [16]. En este diagrama se reflejarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

En la elaboración del diagrama se ha tenido en cuenta la arquitectura del sistema elegida en el apartado 5.2.2, por lo que los componentes se dividen en 3 capas claramente diferenciadas: presentación, negocio y datos.



Los componentes de la capa de presentación ofrecen interfaces a los componentes de la capa de negocio, los cuales utilizan estas interfaces para mostrar al usuario los resultados de sus acciones.

Los componentes de la capa de presentación dependen de los componentes de la capa de negocio, pues son estos los que traducen los eventos de usuario a llamadas de componentes de la capa de datos.

Los componentes de la capa de negocio dependen del componente de la capa de datos al ser este el que realmente lleva a cabo las tareas requeridas por el usuario. El único componente existente en la capa de datos se encarga de gestionar toda la información que recibe y devolver la solicitada por la capa de negocio.

5.2.3.3. Responsabilidades

Cada uno de los componentes tiene varias responsabilidades, que pueden clasificarse en específicas y generales. La descripción completa de los componentes viene dada por estas responsabilidades, junto con el diagrama de componentes especificado anteriormente.

5.2.3.3.1. Responsabilidades Generales

Las responsabilidades generales se refieren a las responsabilidades que únicamente dependen de la capa de la arquitectura en la que se encuentran:

Los componentes de la capa Presentación tienen la responsabilidad de ofrecer al usuario una interfaz gráfica que le permita acceder a los servicios de la capa de negocio y visualizar la información que proporcionen dichos servicios.

Los componentes de la capa Negocio tienen la responsabilidad de ejecutar los servicios solicitados por la capa de presentación, enviar los resultados de dichos servicios a la misma, y acceder a la capa de datos cuando sea requerido para llevar a cabo un servicio.

Los componentes de la capa Datos tienen la responsabilidad de realizar operaciones (consulta, inserción, modificación, borrado) sobre la base de datos, según sean solicitadas por la capa de Negocio y enviar en su caso los resultados a dicha capa.

5.2.3.3.2. Responsabilidades Específicas

Las responsabilidades específicas se refieren a las funcionalidades que ofrecen los componentes, y que relacionan lo indicado en cada capa del diagrama de componentes con los requisitos funcionales y no funcionales del software.

La relación entre los componentes de cada capa arquitectónica y requisitos del software se detalla a continuación, mostrando los principales componentes. Se han agrupado los componentes de presentación en uno sólo para facilitar las relaciones.

- Componentes de Presentación: RF-05, RF-06, RF-12, RF-14, RF-15, RF-16, RNF-01.
- Gestión de usuarios: RF-01, RF-02, RF-03, RF-04, RF-05, RF-14 y RF-17.
- Gestión de preguntas: RF-07, RF-09, RF-15 y RF-16.
- Gestión de respuestas: RF-08, RF-09 y RF-15.
- Gestión de calificaciones: RF-10, RF-11, RF-12 y RF-13.
- Gestión de opciones: RF-02, RF-03, RF-05, RF-06, RF-09, RF-11, RF-12 y RF-13.
- Gestión de base de datos: RF-01, RF-02, RF-03, RF-05, RF-07, RF-08, RF-09, RF-10 y RF-12.

Los requisitos no funcionales RNF-02, RNF-03, RNF-04 y RNF-05 no aportan ninguna funcionalidad pero afectan a todos los componentes de manera indirecta en la fase de implementación.

5.2.3.3.3. Matriz de Rastreabilidad

La matriz de rastreabilidad proporciona una visión rápida y sencilla de las relaciones que existen entre los requisitos funcionales-no funcionales y componentes del sistema. Además, demuestra que en el diseño se han contemplado todas las necesidades presentadas inicialmente.

	Componentes Presentación	Gestión Usuarios	Gestión Preguntas	Gestión Respuestas	Gestión Calificaciones	Gestión Opciones	Gestión Base Datos
RF-01		⊖					⊖
RF-02		⊖				⊖	⊖
RF-03		⊖				⊖	⊖
RF-04		⊖					
RF-05	⊖	⊖				⊖	⊖
RF-06	⊖					⊖	
RF-07			⊖				⊖
RF-08				⊖			⊖
RF-09			⊖	⊖		⊖	⊖
RF-10					⊖		⊖
RF-11					⊖	⊖	
RF-12	⊖				⊖	⊖	⊖
RF-13					⊖	⊖	
RF-14		⊖					
RF-15	⊖		⊖	⊖			
RF-16	⊖		⊖				
RF-17		⊖					
RNF-01	⊖						
RNF-02	⊖	⊖	⊖	⊖	⊖	⊖	⊖
RNF-03	⊖	⊖	⊖	⊖	⊖	⊖	⊖
RNF-04	⊖	⊖	⊖	⊖	⊖	⊖	⊖
RNF-05	⊖	⊖	⊖	⊖	⊖	⊖	⊖

6. Implementación y Pruebas

Tras finalizar la fase de diseño, comienza la fase de implementación y pruebas. Es prioritario en esta fase tener en cuenta los requisitos del software, el esquema relacional y el diseño de componentes detallado en secciones anteriores. Esta fase se compone de la implementación y pruebas de la base de datos y de la aplicación.

6.1. Base de datos





Es importante contar con un esquema relacional y un buen SGBD para realizar la implementación y pruebas de la base de datos. El esquema relacional se encuentra en la fase de diseño anteriormente detallada, y el SGBD se debe elegir en esta fase.

6.1.1. Selección de SGBD

La elección del SGBD debe llevarse a cabo teniendo en cuenta los requisitos de la aplicación y las necesidades del proyecto. Las características que debe tener el SGBD son las siguientes:

- El SGBD debe ser escalable, y por tanto, debe permitir modificar bases de datos existentes, añadir índices para acelerar consultas y sufrir un impacto mínimo en futuras ampliaciones (*RNF-05*).
- El SGBD debe ser sencillo, ya que no se manejarán grandes volúmenes de datos.
- El SGBD debe estar disponible tanto en Linux como en Windows (*RNF-02*).
- Por limitaciones económicas del proyecto, el SGBD debe poder utilizarse sin ningún tipo de coste.
- El SGBD debe ser compatible con la tecnología que se utilice en la implementación del sistema, que en nuestro caso es JSP.

Los SGBD que pueden amoldarse más a estas características son Oracle, PostgreSQL, MySQL, SQLServer y Access. Oracle no es gratuita, por lo que se puede descartar. En la siguiente tabla se realizará un estudio comparativo del resto de sistemas a partir de las principales características que posee un SGBD [16].

	ACCESS	SQL Server	MYSQL	PostgreSQL
Plataforma				
Licencia	Microsoft	Microsoft	GPL	BSD
Estabilidad	Poca	Mucha	Mucha	Mucha
Lenguaje utilizado popularmente	Visual Basic / C++	.NET	PHP	Java
Ambiente	Local	Servidor	Servidor / Internet	Web / Internet
Capacidad de almacenamiento	Limitada	Media	Amplia	Amplia
Cantidad de usuarios en tiempo real	Limitada	Limitada	Amplia	Amplia
Nivel de proyectos	Pequeños	Medios	Avanzados	Avanzados
Consistencia	Mala	Media	Buena	Buena
Velocidad	---	Alta	Alta	---
Integridad Referencial	Alta	Alta	Media	Alta
Tamaño Máximo	Bajo	Medio	Alto	Alto
Seguridad	Mala	Media	Buena	Media

Se pueden descartar Access y SQL Server, que sólo están disponibles para Windows. PostgreSQL es de 2 a 3 veces más lento que MySQL y consume más recursos.

Teniendo en cuenta principalmente que necesitamos un SGBD gratuito y que funcione en Windows y Linux, resulta evidente que el SGBD que mejor se adapta a las necesidades del proyecto es MySQL.

6.1.2. Implementación y Pruebas

El paso siguiente a la elección del SGBD es la implementación de la base de datos. Al basarse en el esquema relacional realizado en la fase de diseño, resulta sencillo de implementar. Las acciones que se han llevado a cabo son: crear tablas, definir claves primarias, alternativas y ajenas, restringir el valor de algunos atributos y el tipo de borrado. La implementación de dominios, comprobaciones asociadas a estos dominios y aserciones se realizaron sobre la propia base de datos y directamente sobre el código de la aplicación.

Durante la implementación se han realizado pruebas unitarias (inserciones, modificaciones, borrados y consultas) para comprobar el funcionamiento de la base de datos. Estas pruebas controlaban el uso de claves primarias, secundarias y ajenas, el valor de atributos con restricción, el borrado y modificación en cascada, etc. Al finalizar la implementación se probó cada una de las sentencias SQL necesarias para el funcionamiento de la aplicación.

Entre las pruebas realizadas, la más importante fue quizá la encargada de calcular la nota media de los alumnos a partir de otras notas medias ponderadas.

Para realizar estas mediciones se ha creado la base de datos y realizado inserciones en todas las tablas. En total se han insertado para realizar pruebas 5 preguntas, 20 respuestas y 30 calificaciones. A continuación, se han medido los tiempos que tardan las diferentes consultas en un ordenador con prestaciones medias para buscar una optimización si el tiempo obtenido fuera superior al esperado, pero ninguna de las consultas supone un gasto de tiempo demasiado grande, son consultas bastante simples, por lo que no es necesario optimizarlas.

6.2. Aplicación Software

La implementación y pruebas de la aplicación software se han llevado a cabo partiendo del diseño de la arquitectura y de componentes realizados en la fase de diseño.

6.2.1. Selección del Lenguaje de Programación

Para poder implementar se debe seleccionar el lenguaje de programación que más se adapte a las necesidades del proyecto.

Según el requisito *RNF-02*, la aplicación debe poder visualizarse en plataformas Windows y Linux, por tanto, el lenguaje elegido debe ser portable. Además, debido a la planificación y disponibilidad de recursos, es necesario elegir un lenguaje cómodo para el programador.

Teniendo en cuenta las consideraciones anteriores, se decidió que la tecnología de programación más conveniente para implementar la aplicación software era JSP, que permite generar contenido dinámico para web y utilizar código Java mediante scripts [17].

6.2.2. Instalación del software inicial

Antes de comenzar con la programación, ha sido necesaria la instalación del software base para poder implementar la aplicación. La idea inicial fue utilizar la herramienta Eclipse Helios, en la cual había que incluir los plugins para conectar con la base de datos, configurar MySQL e instalar Apache Tomcat, pero resultó difícil encajar todas las piezas y lograr tener un software estable que funcionara tanto en Windows como en Linux, por lo que finalmente se decidió utilizar Netbeans 7.2, que incluye por defecto los plugins de conexión con la base de datos y el servidor, sin necesidad de realizar otras configuraciones. El único plugin a añadir es el necesario para enviar emails (mail.jar).

El desarrollo de la aplicación se realizó en una máquina con SO Windows, mientras que el servidor proporcionado para la implantación tenía un SO Linux. Esto fue un inconveniente a la hora de instalar el software base, ya que existen versiones diferentes para ambos sistemas operativos. Finalmente, debido a imprevistos ajenos al proyecto, la aplicación tuvo que volver a migrarse a Windows.

6.2.3. Estructuración en Incrementos

La fase de implementación y pruebas se ha realizado en incrementos. Basándose en el diseño de componentes, se han decidido dividir el proyecto en 5 incrementos, cada uno de los cuales tendrá un objetivo y contendrá una o varias funcionalidades del sistema, que se implementarán y probarán de forma independiente al finalizar cada uno de los incrementos.

El número de incrementos coincide con el número de componentes que posee la capa arquitectónica de negocio. Cada uno de estos incrementos incluye las funcionalidades

indicadas en el apartado 5.2.3.3.2 y son: gestión de usuarios, gestión de preguntas, gestión de respuestas, gestión de opciones y gestión de calificaciones.

Todos los componentes, excepto los de presentación y gestión de base de datos, deben ser completamente implementados y probados dentro de su incremento.

6.2.4. Implementación de la Arquitectura

La implementación de la arquitectura en 3 capas en entornos Java es una práctica común, por lo que está bastante documentada. Antes de comenzar a implementar el sistema, se estudió la mejor forma de implementar esta arquitectura. Los resultados del estudio se muestran a continuación:

- El usuario recibe los datos e información de la interfaz de forma indirecta a través del servidor.
- Las capas de negocio y datos pueden residir en el mismo ordenador, aunque lo común es que se encuentren en ordenadores diferentes.
- Si el usuario navega por las páginas debe ser controlado en la capa de presentación.
- Sólo existe una capa de presentación, negocio y datos.
- La capa de negocio debe manejar todas las páginas de la capa de presentación.
- La capa de presentación no puede comunicarse con la capa de datos sin utilizar la capa de negocio.
- El patrón de diseño a seguir es MVC (Modelo Vista Controlador), muy utilizado en aplicaciones web, y que se corresponde con la arquitectura 3 capas ya que separa los datos, interfaz de usuario y la lógica de negocio en tres componentes distintos.

6.2.5. Implementación del Sistema

Cada implementación de un incremento del sistema supone una serie de tareas que se detallan a continuación:

- **Diseño detallado:** antes de la implementación, se ha realizado por cada incremento un diagrama de interacción informal para codificarlo sobre una base sólida. La intención principal ha sido crear entidades genéricas con el fin de que el mantenimiento y evolución del sistema sea lo más sencillo posible.

Esta tarea podría ser una fase en el desarrollo del proyecto, pero por su carácter informal se posiciona como una tarea de implementación.

- **Diseño IU:** las interfaces de usuario de cada incremento se han esbozado antes de su implementación, y tras ser validadas por el usuario se han codificado.
- **Codificación:** tarea de programación.
- **Documentación:** el código de programación aparece comentado en detalle para ofrecer una evolución y mantenimiento fácil. Esto ofrece una documentación amplia y de calidad.

Para ofrecer una visión general del tamaño y organización del código desarrollado, se resume a continuación el número de cifras más representativas:

Concepto	Cifra
Líneas de código	~ 7000
Páginas JSP	35
Paquetes	16

6.2.6. Pruebas

Las pruebas llevadas a cabo se dividen en 3 tipos: unitarias, de usabilidad y de validación. Estas pruebas se han realizado en función de los incrementos detallados en el apartado 6.2.3.

- **Pruebas unitarias:** Por cada funcionalidad implementada se ha ejecutado una prueba de la misma para comprobar su correcto funcionamiento.
- **Pruebas de usabilidad:** Se han utilizado 5 usuarios para realizar las pruebas de cada incremento: dos compañeros de trabajo familiarizados con el uso aplicaciones web, dos estudiantes de carreras universitarias no tecnológicas y un familiar que no está acostumbrado a utilizar este tipo de aplicaciones.

Para efectuar estas pruebas, cuya duración ha sido de unos 10 minutos, se han realizado formularios (Anexo1) con tareas y cuestiones que los usuarios han cumplimentado. Además, los usuarios tenían libertad de aportar nuevas ideas o mejoras que creyeran oportunas.

Al finalizar cada una de estas pruebas se estudiaban los resultados para comprobar si el incremento era usable, de lo contrario se realizarían los cambios pertinentes y se realizaría después una prueba de validación.

De estas pruebas surgieron ideas interesantes como subrayar las palabras que redirigen a otra pantalla, mostrar mensajes pop-up más descriptivos, añadir campos en algunas tablas para mostrar información que a los usuarios les parecía útil, etc. El Anexo 1 contiene la información detallada de estas pruebas.

- **Pruebas de validación:** Consisten en mostrar al usuario una simulación del incremento a probar. Según la opinión del usuario se consideraba o no cumplido el objetivo del incremento, en caso negativo se modificaba el incremento hasta lograr su aceptación. En este proyecto se obtuvieron pruebas de validación satisfactorias.

Por otro lado, se han realizado validaciones del código CSS y HTML a través del sitio web W3C (<http://validator.w3.org/>) y se han incluido los iconos correspondientes en todas las páginas de la aplicación.

7. Planificación Final y Análisis de Costes

En este apartado se detalla la planificación final y el análisis de coste del proyecto que se ha llevado a cabo.

7.1. Planificación Final

La planificación final varía con respecto a la planificación inicial llevada a cabo al principio del proyecto. Al comienzo del proyecto se realizó una estimación, en la cual se estimaba el tiempo que las diferentes tareas conllevarían. A medida que el proyecto avanzaba, esta planificación iba sufriendo cambios:

La duración de captura de requisitos de software se alargó 4 días más de lo estimado: determinar la funcionalidad que se podría ofrecer en un sistema como el que se ha llevado a cabo fue una ardua tarea, ya que hubo que estudiar y analizar varias aplicaciones semejantes y encontrar puntos a mejorar y novedosos que poder explotar.

El diseño de la base de datos duró 4 días más de lo estimado: el estudio de SGBD y las entidades, tablas y claves necesarias para tratar los datos fue más costoso de lo previsto. Asimismo, se retrasó 7 días la implementación de la base de datos debido a la novedad que suponía este gestor para el programador.

La instalación del software, tanto en la máquina de desarrollo como en el servidor, se retrasó 25 días debido a la dificultad que supuso la búsqueda de las versiones correctas del software y debido también a los problemas que ocasionaba la instalación en distinto orden del mismo. Este problema se solucionó cambiando de Eclipse a Netbeans.

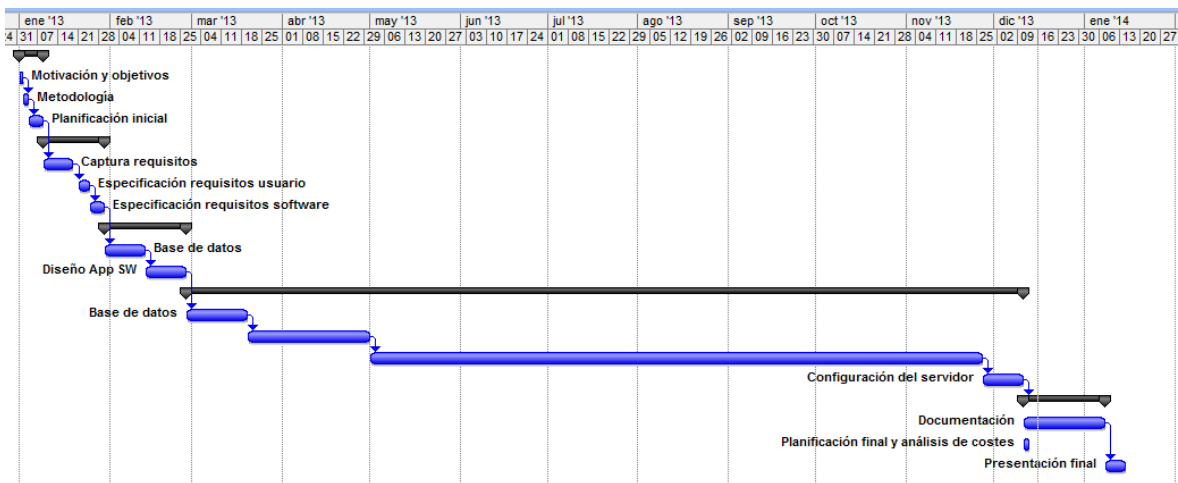
El retraso más importante tuvo lugar en la fase de implementación de la aplicación software, al ser un lenguaje de programación desconocido inicialmente para el programador, la estimación inicial difiere bastante del resultado final. Además, durante la fase de implementación se produjeron varios imprevistos y hubo que rehacer parte de la aplicación. La duración de esta fase difiere en 80 días con respecto a la planificación inicial.

La documentación se retrasó 5 días ya que se ha realizado un estudio profundo de otras aplicaciones similares, una guía de usuario detallada y multitud de pruebas de la aplicación.

Finalmente, la planificación de este proyecto abarca desde el 1 de enero de 2013 hasta el 28 de diciembre de 2013, en total, 271 días, 135 días más de los estimados inicialmente.

A continuación se muestra el diagrama de Gantt con la planificación detallada anteriormente, donde se plasma la duración de cada una de las fases en las que se ha dividido el proyecto:

Nombre de tarea	Duración	Comienzo	Fin
Inicio del proyecto	6 días	mar 01/01/13	mar 08/01/13
Motivación y objetivos	1 día	mar 01/01/13	mar 01/01/13
Metodología	2 días	mié 02/01/13	jue 03/01/13
Planificación inicial	3 días	vie 04/01/13	mar 08/01/13
Análisis	15 días	mié 09/01/13	mar 29/01/13
Captura requisitos	8 días	mié 09/01/13	vie 18/01/13
Especificación requisitos usuario	4 días	lun 21/01/13	jue 24/01/13
Especificación requisitos software	3 días	vie 25/01/13	mar 29/01/13
Diseño	20 días	mié 30/01/13	mar 26/02/13
Base de datos	10 días	mié 30/01/13	mar 12/02/13
Diseño App SW	10 días	mié 13/02/13	mar 26/02/13
Implementación y pruebas	205 días	mié 27/02/13	mar 10/12/13
Base de datos	15 días	mié 27/02/13	mar 19/03/13
Instalación software base	30 días	mié 20/03/13	mar 30/04/13
Aplicación software	150 días	mié 01/05/13	mar 26/11/13
Configuración del servidor	10 días	mié 27/11/13	mar 10/12/13
Memoria final	20 días	mié 11/12/13	mar 07/01/14
Documentación	20 días	mié 11/12/13	mar 07/01/14
Planificación final y análisis de costes	2 días	mié 11/12/13	jue 12/12/13
Presentación final	5 días	mié 08/01/14	mar 14/01/14



7.2. Análisis de Costes

En este apartado se detalla el análisis de costes realizado con la finalidad de valorar este proyecto desde el punto de vista económico.

7.2.1. Recursos Humanos

En este proyecto, se han utilizado los siguientes recursos humanos:

- Un gestor de proyecto que trabaja durante todo el proyecto.
- Un analista/diseñador que trabaja en las fases de análisis y diseño.
- Un diseñador de interfaces de usuario durante la primera parte de la fase de implementación.
- Un programador que trabaja durante las fases de implementación y pruebas.

Los costes derivados de recursos humanos se especifican en la siguiente tabla:

Rol	€/hora	Horas	Coste (€)
Gestor Proyecto	15	1.084	16.260
Analista/Diseñador	12	140	1.680
Diseñador IU	10	50	500
Programador	8	205	1.640
Total			20.080

7.2.2. Equipamiento

Para llevar a cabo este proyecto ha sido necesario el siguiente equipamiento hardware: un ordenador y una impresora.

En cuanto al equipamiento software, ha sido necesario un paquete de Microsoft Office 2010 incluyendo Microsoft Visio y Microsoft Project. El resto de software utilizado es gratuito, por lo que se omitirá.

Los costes asociados al equipamiento se detallan a continuación:

Producto	Coste (€)
Ordenador	1.000
Impresora	100
Microsoft Office	130
Total	1.230

7.2.3. Fungibles

Los gastos en fungibles y material de oficina se especifican en la siguiente tabla:

Producto	Coste (€)
Cartucho Impresora	30
Papel	5
CD	5
Total	40

7.2.4. Otros

En el proyecto ha habido una serie de gastos que no encajan en los apartados anteriores, se especifican en la tabla siguiente:

Producto	Coste (€)
Transporte	100
Luz, teléfono, Internet	400
Total	500

7.2.5. Coste Total

A continuación se detalla el coste total del proyecto:

Producto	Coste (€)
Recursos Humanos	20.080
Equipamiento	1.230
Fungibles	40
Otros	500
Total	21.850

Como de indica en la tabla anterior, el coste total del proyecto asciende a 21.850€. Se debe tener en cuenta que si el proyecto fuera comercial habría incrementos de beneficios, en torno a un 10%, que la empresa desarrolladora cobraría.

8. Conclusiones y Trabajo Futuro

Para finalizar, se ofrecen una serie de conclusiones del proyecto y aspectos que en un futuro se podrían mejorar o funcionalidades que se podrían incorporar a la aplicación.

8.1. Conclusiones

El nuevo Plan Bolonia implantado en las universidades ha provocado que tanto alumnos como profesores tengan más trabajo, ya que se debe llevar un seguimiento continuo del estudio de los alumnos.

Actualmente existen aplicaciones para algunas asignaturas de la universidad: foros, webs donde colgar apuntes, etc. En concreto, la Universidad Carlos III de Madrid ha desarrollado la aplicación “Aula Global”. Esta aplicación permite a los profesores compartir con los alumnos material docente, crear distintas actividades online (cuestionarios, entrega de prácticas, etc.), y además proporciona un medio de comunicación mediante el uso de foros. Sin embargo, no existe ningún modulo que permita al alumno formular preguntas ni responder a preguntas formuladas por otros compañeros, y que esta participación pueda ser evaluada de forma automática, facilitando así la tarea de evaluación del profesor. De este modo, con la aplicación desarrollada en este proyecto, por un lado se fomenta la participación del alumno en su proceso de aprendizaje y se reduce la carga de trabajo del docente en el proceso de evaluación. Por estos motivos, el proyecto llevado a cabo resulta novedoso y muy a tener en cuenta para el proceso de evaluación de las asignaturas con el nuevo plan, pues ha surgido una necesidad que debe ser cubierta.

En relación al desarrollo del proyecto, debo decir que ha sido muy interesante y me ha ayudado a aprender un conjunto de tecnologías (Java, Javascript, Html, MySQL) que no había visto antes y que son muy utilizadas en el mundo laboral.

Tras el estudio de los resultados obtenidos en las pruebas de usabilidad con usuarios reales, se puede definir como buena la utilidad que aporta la aplicación desarrollada, la cual ha sido catalogada como agradable a la vista, intuitiva y fácil de utilizar. Espero que la aplicación desarrollada en este proyecto suponga una gran ayuda para los profesores y resulte muy útil tanto a profesores como a alumnos.

Para finalizar, se ha intentado implementar un sistema con la mayor escalabilidad posible, haciendo que pueda utilizarse en varias plataformas y para diversidad de materias. Esta escalabilidad se debe al uso de la arquitectura en 3 capas; la presentación se ha implementado en páginas Html y el negocio en páginas JSP, por lo que si se desea añadir funcionalidades nuevas, lo único que habrá que hacer será modificar el Html para añadir el botón, enlace, tabla o elemento que corresponda y añadir una nueva página JSP por cada funcionalidad que se desee incluir. Es decir, no se debe modificar el código que existe actualmente, una nueva funcionalidad implica un nuevo módulo. Por esta razón, añadir nuevas funcionalidades y adaptar el sistema para gestionar otras asignaturas son procedimientos que se podrían realizar de forma sencilla.

8.2. Trabajo Futuro

A continuación se proponen algunas ideas para ampliar y mejorar la versión implementada:

La principal mejora sería plantear la aplicación para que pudiera integrar varias asignaturas, de manera que hubiera varios temas de cada asignatura y los alumnos pudieran plantear sus dudas y responder a preguntas. De hecho, sería muy interesante que la aplicación desarrollada fuera integrada como un módulo más de Aula Global, para que pudiera ser utilizada por toda la comunidad docente y estudiantes de la UC3M. En relación con esta idea, otra mejora interesante sería permitir que hubiera varios profesores y cada uno de ellos tuviera asociados unos alumnos, de manera que un profesor sólo pudiera ver las respuestas dadas por sus alumnos y calificarlas.

Una de las ideas que se planteó al inicio del proyecto fue que los alumnos pudieran responder adjuntando algún tipo de fichero, un PDF por ejemplo, respondiendo a la pregunta con más detalle, pero por imprevistos ajenos al proyecto y debido a que el desarrollo finalmente no se pudo realizar en un servidor de la universidad, resultó imposible utilizar un servidor FTP para guardar estos ficheros. Como ampliación a futuro sería sencillo de implementar y una funcionalidad muy útil.

Otras ideas de ampliación para el perfil profesor serían, por ejemplo, poder consultar las preguntas y respuestas agrupadas por autor (actualmente se consultan a nivel de categoría), plantear formularios con preguntas de teoría que pudieran proporcionar una nota automáticamente al rellenarlos o crear un apartado para colgar tutoriales o apuntes para que los alumnos puedan consultarlos. También podría existir la opción de exportar las notas finales de los alumnos a un documento Excel o PDF.

Además, podría añadirse un buscador en la aplicación, para poder realizar búsquedas por una determinada palabra, o información de un usuario concreto.

De cara al uso inicial de la aplicación, podría resultar útil poder realizar una carga masiva de alumnos, ya que actualmente es necesario que se registren uno por uno.

9. Glosario de Términos y Acrónimos

Término/Acrónimo	Definición
Access	Sistema de gestión de bases de datos relacionales para los sistemas operativos Microsoft Windows.
Apache Tomcat	Servidor web con soporte de servlets y JSPs.
Atributo	En bases de datos, un atributo representa una propiedad de interés de una entidad.
CD	Compact Disc - Disco Compacto.
CO	Componente desarrollo software.
DB2	Sistema de gestión de bases de datos de IBM.
Diagrama E/R	Herramienta de modelado de sistemas que representa entidades, los datos almacenados en el sistema y las relaciones entre éstos.
Diagrama Gantt	Herramienta gráfica utilizada para mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo determinado.
DNI	Documento Nacional de Identidad.
Dominio	En bases de datos, son los valores que puede tomar un atributo.
Eclipse	Plataforma de desarrollo basada en Java, que permite construir entornos de desarrollo a partir de componentes conectados.
EDA	Estructura de Datos y Algoritmos. Asignatura de Grado en Ingeniería Informática.
EEES	Espacio Europeo de Educación Superior.
Escala Guttman	Escala que ordena todas las respuestas en base a una sola característica o atributo, presentándose los estímulos de sencillos a más complejos.
Escala Lickert	Escala comúnmente utilizada en cuestionarios que consiste especificar el nivel de acuerdo o desacuerdo con una declaración.

FTP	File Transfer Protocol. Permite transferir archivos locales hacia un servidor web.
Grafo	Unidad abstracta que permite expresar de una forma visualmente muy sencilla y efectiva las relaciones que se dan entre elementos de muy diversa índole.
HTML	Hyper Text Markup Language. Lenguaje con el que se definen las páginas web. Se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que componen una página web.
IP	Internet Protocol. Número que identifica un dispositivo en una red.
IU	Interfaz de Usuario.
JAVA	Lenguaje de programación orientado a objetos.
JDK	Java Development Kit. Software que provee herramientas de desarrollo para la creación de programas en Java.
JRE	Java Runtime Environment. Máquina virtual de Java cuya función es hacer de intermediario entre una aplicación programada en Java y el sistema operativo.
JSP	Java Server Pages. Tecnología Java que permite a los desarrolladores de software generar dinámicamente HTML, XML u otros tipos de documentos, en respuesta al requerimiento de un cliente web.
Linux	Núcleo de sistema operativo de libre distribución Unix para computadoras personales, servidores y estaciones de trabajo.
Matriz de rastreabilidad	Matriz que permite relacionar dos tipos de elementos.
Microsoft Office	Suite de oficina que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios para el sistemas operativo Microsoft Windows.
MySQL	Sistema de gestión de bases de datos relacional, multihilo y multiusuario.
NIU	Número de Identificación de Usuario.
Oracle	Sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation.

Paradigma	Ideologías creativas de los evaluadores. Determinan el pensamiento y comportamiento metodológico de los evaluadores.
Plugin	Módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.
RAM	Random Access Memory. Memoria de ordenador a la que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes.
RF	Requisito de Software Funcional.
RNF	Requisito No Funcional.
SGBD	Sistema Gestor de Bases de Datos.
SO	Sistema Operativo.
SQL	Structured Query Language. Lenguaje de consulta estructurado que permite especificar diversos tipos de operaciones que se realizan sobre bases de datos.
Supuesto semántico	Hecho asumido como cierto a efectos de la planificación del proyecto.
TIC	Tecnologías de Información y Comunicación. Agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, internet y telecomunicaciones.
UML	Unified Modeling Language - Lenguaje Unificado de Modelado Conjunto de herramientas que permiten analizar y diseñar sistemas orientados a objetos.
URL	Localizador de Recurso Uniforme - Uniform Resource Locator. Dirección global de documentos y de otros recursos en la World Wide Web.
Windows	Familia de sistemas operativos desarrollados por Microsoft.

10. Referencias

- [1] <http://www.duoc.cl/cfd/docs/aprendizaje-activo-metodologias-ducativas.pdf>
- [2] <http://www.uab.es/servlet/Satellite/bolonia/estudiantes/-que-debes-esperar-1231747994976.html>
- [3] <http://www.queesbolonia.gob.es/queesbolonia/bolonia-para-ti/profesor/mas-alla-del-examen-hacia-la-evaluacion-continua.html>
- [4] <http://el-eees-no-es-de-broma.blogspot.com/>
- [5] www.elpais.com/artículo/sociedad/Recortes/burocracia/lastran/Bolonia
- [6] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>
- [7] <https://apps.lab.inf.uc3m.es/foro>
- [8] <http://delegacion.uc3m.es/blog/>
- [9] <http://es.answers.yahoo.com/>
- [10] <http://www.osqa.net/>
- [11] <http://moodle.org/>
- [12] <http://www.efrontlearning.net/>
- [13] <http://www.webquest.es/>
- [14] Concepción y Diseño de Bases de Datos: Del Modelo E/R al Modelo Relacional. A.de Miguel, 1993.
- [15] <http://metodologiasdesistemas.blogspot.com.es/2007/05/diseo-en-3-capas-fisicas-lgicas-es.html>
- [16] <http://www.slideshare.net/ManuelMirandaBuenabad/cuadro-comparativo-sgbd>
- [17] http://www.programacionfacil.com/programacion/manual_java_jsp

Anexo 1 – Pruebas de Usabilidad

Antes de comenzar con las pruebas, es importante decidir qué método de evaluación se va a utilizar para comprobar la usabilidad de la aplicación web, para ello nos basaremos en tres procesos básicos: paradigma, método de evaluación y técnica de experimentación.

Existen multitud de paradigmas en el proceso de evaluación de sistemas, el utilizado en este caso ha sido *Usability Testing*, puesto que analiza el uso que hacen los usuarios de la aplicación a la hora de realizar tareas típicas y sólo requiere el control del evaluador.

El método de evaluación usado es *Experimentación o Testing*, ya que es el que mejor se ajusta a nuestras necesidades, debido a que es aquel en el que los evaluadores emplean los resultados obtenidos con usuarios representativos utilizando el sistema en tareas típicas.

La técnica de experimentación que se ha empleado es *Protocolo de Pensamiento en Voz Alta*, ya que nos permite obtener consideraciones particulares de los usuarios del sistema mientras estos desarrollan una serie de tareas de las cuales obtendremos una información cuantificable; de esta manera, cubriremos con garantías los objetivos de la evaluación (mejorar la eficiencia de la navegación de los usuarios por la aplicación web, cuantificar el agrado con el diseño de la web y comprobar la funcionalidad desarrollada).

Los resultados de la evaluación se registrarán en la tabla que se muestra a continuación. Existirá una tabla por tarea a realizar por un usuario, y se ajustarán los datos a los fijados por el evaluador en tiempo y número de clics:

Tarea	Excelente	Aceptable	Inaceptable
Tiempo para la tarea	2 minutos	2–4 minutos	> 4 minutos
Número de errores cometidos en la navegación	1	2 - 3	> 3
Número de pulsaciones al botón “Volver” por desorientación del usuario	0	>1	> 2
Comentarios realizados por el usuario durante la prueba	El formato a seguir en comentarios es: Número de comentario / Comentario / Tiempo		

Además de la realización de estas tareas, se solicita a los usuarios que participan en la evaluación, que expresen en voz alta y en presencia del desarrollador de la aplicación, los pensamientos, opiniones y sensaciones que vayan experimentando durante su interacción con el sistema. El desarrollador anota estos comentarios en la tabla anterior durante el proceso de la evaluación. Las tareas realizadas por los usuarios en cada uno de los incrementos han sido las siguientes:

- ☒ Gestión de usuarios: Dar de alta un usuario, validar usuarios y borrar usuarios.
- ☒ Gestión de preguntas: Ver las preguntas de un tema en concreto, crear una nueva pregunta, borrar y denunciar pregunta.
- ☒ Gestión de respuestas: Ver las respuestas de una pregunta concreta, crear una nueva respuesta, borrar y denunciar respuesta.
- ☒ Gestión de opciones: Cambiar de contraseña, ver un listado de “mis notas” y ver el listado de alumnos.
- ☒ Gestión de calificaciones: Votar respuesta y calcular nota final de alumnos.

Finalmente, cada participante cumplimenta un formulario, del que se identifican factores subjetivos importantes para la aplicación web, como son la facilidad de uso del producto, la importancia de la funcionalidad desarrollada, etc. Se muestra a continuación el formulario final de evaluación de la aplicación:

Pregunta	Respuesta/Comentarios
¿La navegación por la aplicación le parece lógica e intuitiva?	
¿Echa en falta alguna funcionalidad?	
¿Le agrada la combinación de colores y formato empleados?	
¿Reubicaría algún elemento de la web en otro lugar?	
¿Ha utilizado alguna vez una aplicación similar? Si es así, ¿cuál?	
¿Se le ocurre alguna mejora que incorporaría?	
¿Recomendaría esta aplicación?	
¿Se encuentra satisfecho con la aplicación que acaba de probar?	

Resultados obtenidos en las pruebas

A continuación se detallan los resultados obtenidos en las pruebas anteriormente descritas. Estas pruebas han sido realizadas por 5 usuarios: dos compañeros de trabajo familiarizados con el uso aplicaciones web, dos estudiantes de carreras universitarias no tecnológicas y un familiar que no está acostumbrado a utilizar este tipo de aplicaciones.

En la siguiente tabla se muestran los resultados obtenidos de media por los usuarios al realizar varias tareas en cuanto a tiempo, errores de navegación y número de pulsaciones al botón Volver.

Perfil	Tarea	Tiempo	Errores	Volver
	Registrarse	Excelente	Excelente	Excelente
Profesor	Validar usuario	Excelente	Excelente	Excelente
	Borrar usuario	Excelente	Excelente	Excelente
	Calcular notas finales	Aceptable	Aceptable	Excelente
	Ver preguntas de un tema	Excelente	Aceptable	Excelente
	Crear una nueva pregunta	Aceptable	Excelente	Excelente
	Ver respuestas	Excelente	Aceptable	Aceptable
	Crear un comentario	Excelente	Excelente	Excelente
	Denunciar pregunta	Excelente	Excelente	Excelente
	Votar respuesta	Excelente	Aceptable	Excelente
Alumno	Ver notas	Excelente	Excelente	Excelente
	Cambiar contraseña	Excelente	Excelente	Excelente
	Crear una nueva pregunta	Excelente	Excelente	Excelente
	Crear una nueva respuesta	Excelente	Excelente	Excelente
	Votar una respuesta	Excelente	Excelente	Excelente

Administrador	Borrar usuarios	Excelente	Excelente	Excelente
	Borrar preguntas	Excelente	Excelente	Excelente
	Borrar respuestas	Excelente	Excelente	Excelente

Sobre las preguntas del formulario, los usuarios respondieron que se trata de una aplicación con una navegación lógica e intuitiva, fácil de utilizar, con una interfaz gráfica atractiva, y en general, su grado de satisfacción fue bueno.

De estas pruebas surgieron algunas ideas que se utilizaron en la aplicación y otras que se han planteado como posibles ampliaciones futuras en el correspondiente apartado de la memoria. A continuación se enumeran algunas:

- Subrayar las palabras que redirigen a otra pantalla.
- Mostrar mensajes pop-up más descriptivos.
- Añadir campos en algunas tablas para mostrar más información.
- Reubicación de varios botones.
- Adjuntar documentos en las respuestas.
- Añadir botones “Volver” en la propia aplicación en lugar de usar los del navegador.
- Añadir un buscador de preguntas y respuestas por palabras por si los alumnos tienen problemas con algún concepto determinado.
- Consultar las preguntas y respuestas ordenadas por autores.

Anexo 2 – Manual de Instalación

Para implementar la aplicación web desarrollada en este proyecto ha sido necesario instalar y configurar una serie de herramientas. Los pasos seguidos han sido los siguientes:

1º - Descargar **Netbeans 7.2.1**.

<https://netbeans.org/downloads/7.2.1/>

Descargar el paquete Java EE que incluye los servidores GlassFish y Apache Tomcat y el driver para conectar con la base de datos.

2º - Descargar e Instalar **Apache Tomcat 7.0.47**

<http://tomcat.apache.org/download-70.cgi>

3º - Descargar el driver **mail-5.1.0.jar** necesario para poder enviar emails desde la aplicación. Copiar el fichero en la carpeta src/java del espacio de trabajo.

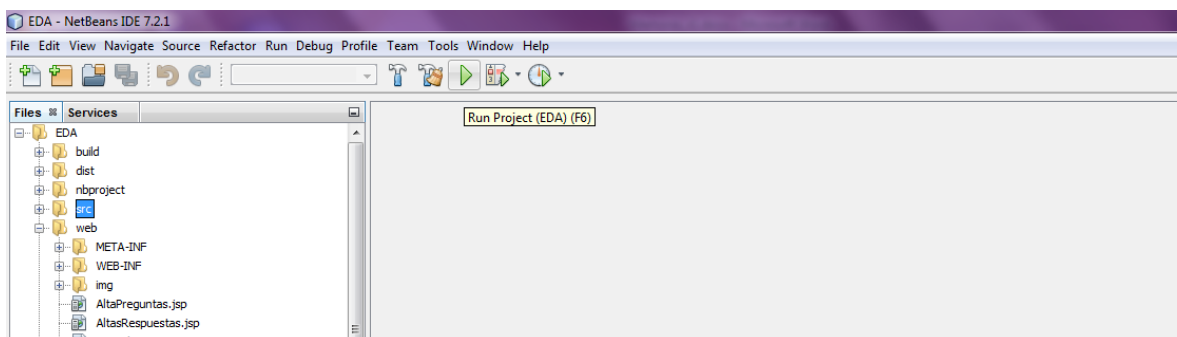
<http://www.java2s.com/Code/Jar/m/Downloadmailjar.htm>

4º - Descargar e instalar **MySQL 6.0** para administrar la base de datos.

<http://mysql.softonic.com/descargar>

5º - Ejecutar en MySQL el script para crear la BD y tablas para la aplicación (\. ruta scriptEDA.txt).

6º - Ejecutar Netbeans. La primera vez solicita la ruta donde se encuentra el espacio de trabajo. Para arrancar la aplicación se pulsa el botón *Run* marcado en la siguiente imagen.

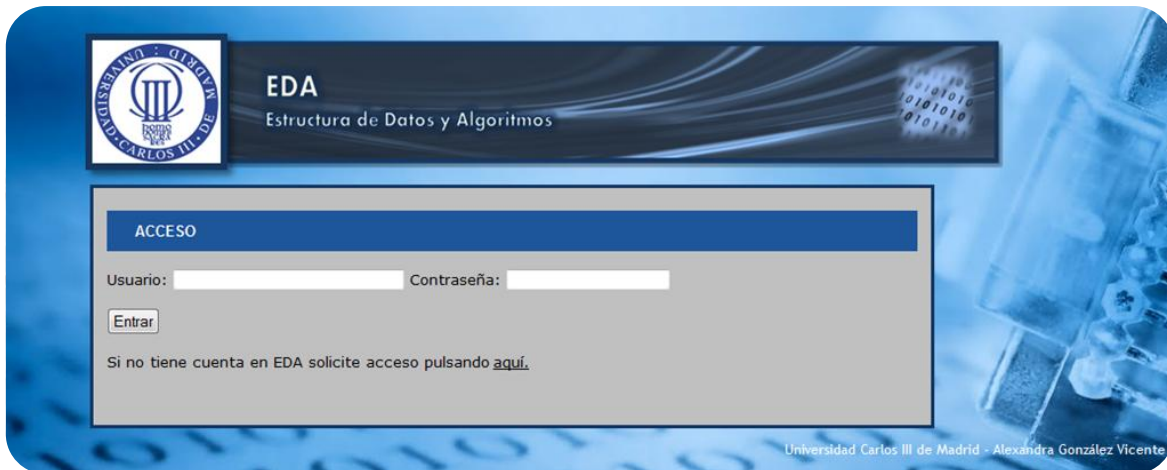


Al finalizar esta instalación en la máquina servidor, la aplicación ya estaría disponible para que los usuarios puedan ejecutarla a través de la dirección <http://IP:8080/EDA/index.jsp>, donde la IP varía en función de la máquina donde esté instalada.

Anexo 3 – Manual de Usuario

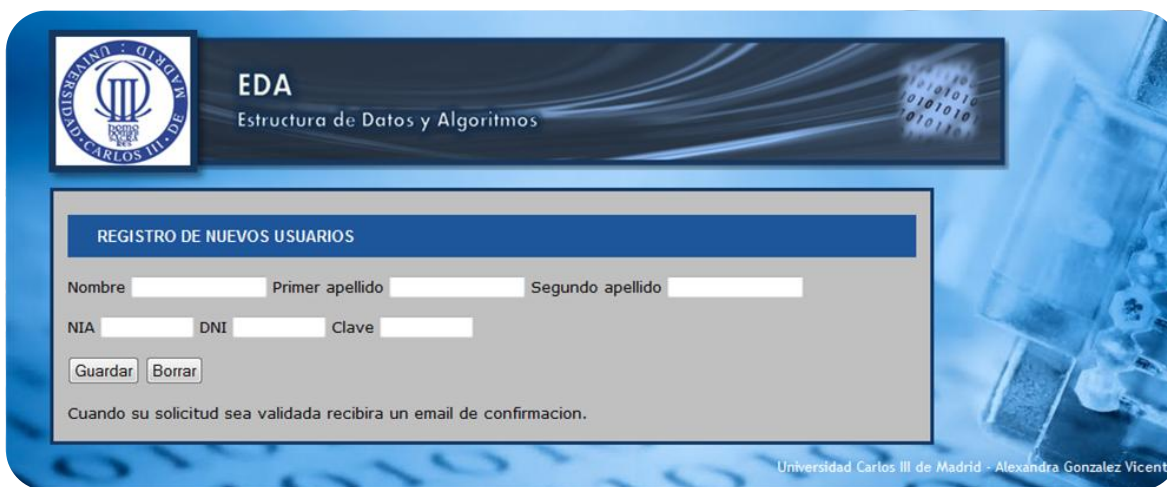
Se trata de una aplicación web, por lo que no requiere instalación. Se accederá a través de la dirección <http://IP:8080/EDA/index.jsp>, donde la IP varía en función de la máquina donde esté instalada.

Existen tres perfiles: alumno, profesor y administrador. En la pantalla inicial para acceder a la aplicación se solicita el usuario (NIU) y la contraseña y se pulsa el botón “Entrar”.



The screenshot shows the login interface of the EDA (Estructura de Datos y Algoritmos) application. At the top left is the logo of the Universidad Carlos III de Madrid. To its right, the text "EDA Estructura de Datos y Algoritmos" is displayed. Below this, there is a section titled "ACCESO" (Access) with a light blue header. Inside this section, there are two input fields: "Usuario:" and "Contraseña:". Below the "Usuario:" field is a button labeled "Entrar". At the bottom of the "ACCESO" section, there is a link that says "Si no tiene cuenta en EDA solicite acceso pulsando [aquí](#)." The background of the page features a blue gradient with binary code and a computer keyboard.

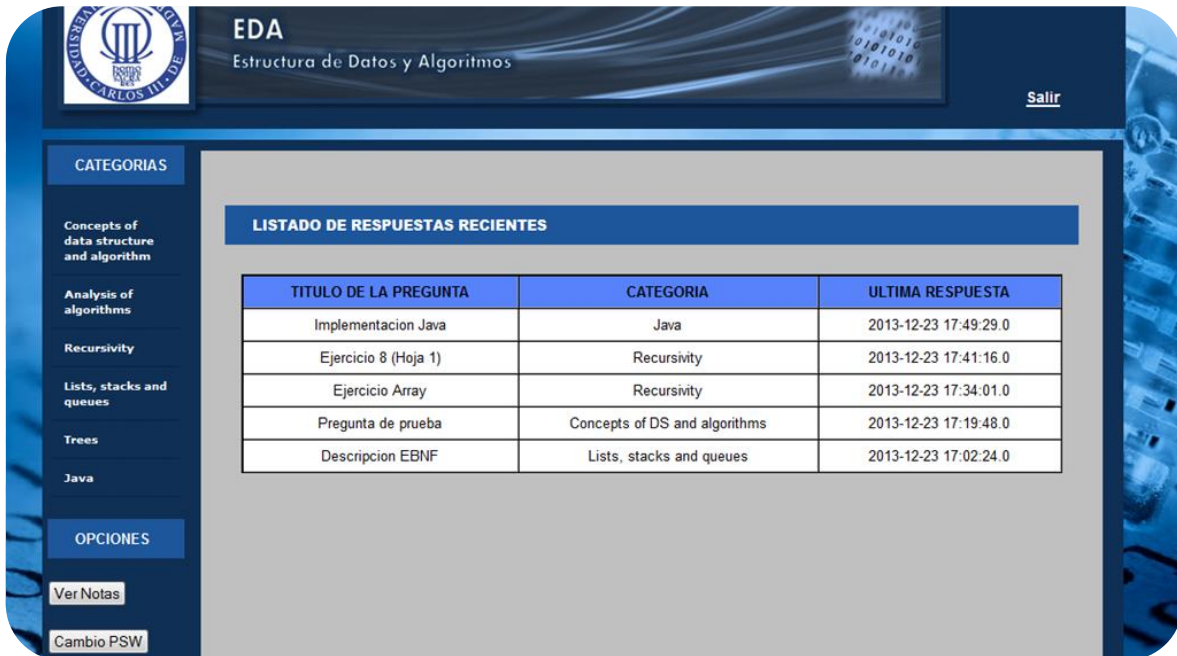
Para darse de alta en el sistema se accede a la página de registro a través del enlace que aparece en la parte inferior de la pantalla anterior, sobre la palabra “aquí”. En esta página se rellenará un formulario con los datos del nuevo usuario. Para acceder al sistema será necesario que el usuario con perfil profesor valide la solicitud de registro, tal y como se detalla a continuación en el perfil del profesor. La página de registro es la siguiente:



The screenshot shows the registration interface of the EDA application. At the top left is the logo of the Universidad Carlos III de Madrid. To its right, the text "EDA Estructura de Datos y Algoritmos" is displayed. Below this, there is a section titled "REGISTRO DE NUEVOS USUARIOS" (Registration of New Users) with a light blue header. Inside this section, there are several input fields: "Nombre", "Primer apellido", "Segundo apellido", "NIA", "DNI", and "Clave". Below the "Nombre" field is a button labeled "Guardar" and a button labeled "Borrar". At the bottom of the "REGISTRO DE NUEVOS USUARIOS" section, there is a note that says "Cuando su solicitud sea validada recibira un email de confirmacion." The background of the page features a blue gradient with binary code and a computer keyboard.

Perfil Alumno

La pantalla principal del perfil alumno muestra una tabla con las 5 preguntas que se han respondido más recientemente. Se puede acceder al detalle de cada pregunta pulsando sobre el título de la misma.



EDA
Estructura de Datos y Algoritmos

[Salir](#)

CATEGORIAS

- Concepts of data structure and algorithm
- Analysis of algorithms
- Recursivity
- Lists, stacks and queues
- Trees
- Java

OPCIONES

- Ver Notas
- Cambio PSW

LISTADO DE RESPUESTAS RECIENTES

TITULO DE LA PREGUNTA	CATEGORIA	ULTIMA RESPUESTA
Implementacion Java	Java	2013-12-23 17:49:29.0
Ejercicio 8 (Hoja 1)	Recursivity	2013-12-23 17:41:16.0
Ejercicio Array	Recursivity	2013-12-23 17:34:01.0
Pregunta de prueba	Concepts of DS and algorithms	2013-12-23 17:19:48.0
Descripcion EBNF	Lists, stacks and queues	2013-12-23 17:02:24.0

Como puede apreciarse en la imagen, la pantalla está compuesta por:

- Una serie de categorías que se corresponden con los temas de la asignatura. Al pulsar sobre cada una de las categorías, navega a la lista de preguntas relacionadas con cada una de ellas.



➤ Las opciones propias del perfil:

- Cambio PSW, donde permite al alumno modificar la contraseña de acceso, para ello el alumno debe introducir la contraseña actual, repetir la nueva contraseña y pulsar el botón “Guardar”. También contiene un botón “Borrar” que limpia los campos.

- Ver notas, donde el alumno puede visualizar las notas recibidas en cada respuesta aportada.

Pregunta	Nota Profesor	Nota Alumno	Nota Resto
Pregunta de prueba	0	0	4,00
Descripcion EBNF	5	4	3,00
Ejercicio 8 (Hoja 1)	0	0	0,00

IMPORTANTE: Estas notas no son definitivas. La nota final la recibirá en su correo.

Pulsando sobre una categoría, se visualizarán las preguntas propuestas sobre el tema elegido y aparecerá un área de texto donde poder proponer nuevas preguntas. Para insertar una nueva pregunta es necesario introducir un título, el texto y pulsar el botón “Guardar”. También existe un botón “Borrar” que limpia los campos. A continuación se muestra un ejemplo donde en la primera imagen puede verse la tabla de preguntas vacía

y se rellenan los campos para crear una nueva pregunta, y en la segunda imagen aparece creada en el listado.

The screenshot shows the 'Insertar una nueva pregunta' (Insert a new question) form. On the left is a sidebar with 'CATEGORIAS' (Categories) and 'OPCIONES' (Options). The main area has a header 'Lists, stacks and queues'. Below it is a table with columns 'Titulo de la Pregunta', 'Autor', and 'Ultima Respuesta'. The 'Titulo de la Pregunta' field is filled with 'Descripcion EBNF'. The 'Autor' field is empty. The 'Ultima Respuesta' field is empty. Below the table is a section 'Insertar una nueva pregunta:' with a 'Titulo:' label and a text input field containing 'Descripcion EBNF'. Below this is a large text area with the following text: 'Una cola circular es una estructura acotada, cuya definición coincide con la de cola, salvo que se incluye el tamaño máximo que ésta puede alcanzar y que el último elemento le sucede el primero. Realice la descripción EBNF de la estructura.' At the bottom are 'Guardar' (Save) and 'Borrar' (Delete) buttons.

The screenshot shows the question list after saving the new question. The table now has one row with the title 'Descripcion EBNF', author '100011111', and last response '--'. The 'Insertar una nueva pregunta:' section is still visible, but the 'Titulo:' field is empty and the text area contains 'Texto...'.

Una vez en esta pantalla, si se pulsa sobre el título de la pregunta, se navega al listado de las respuestas y comentarios del profesor. Además, al igual que en la pantalla de preguntas, existe un área de texto donde poder añadir una respuesta (perfil alumno) o comentario (perfil profesor). Para añadir una nueva respuesta es necesario introducir el texto y pulsar el botón “Guardar”. También existe un botón “Borrar” para limpiar el texto.

Pregunta

Descripcion EBNF

Una cola circular es una estructura acotada, cuya definición coincide con la de cola, salvo que se incluye el tamaño máximo que ésta puede alcanzar y que el último elemento le sucede el primero. Realice la descripción EBNF de la estructura.

Respuestas

NIU	Nombre y Apellidos	Fecha	Nota Media
100033333	Pedro Calvo Martinez	2013-12-23 17:02:24.0	3.00

Comentarios

TEXTOS FECHA

Insertar una nueva Respuesta:

```
<cola> ::= <frente> + <final> + {<nodo>} + << tamaño>> <frente> ::=
<enlace> <final> ::= <enlace> <enlace> ::= << índice>> // donde
índice=0.. tamaño-1 <nodo> ::= <información> + <enlace>
<información> ::= << dato>>{<< dato>>}
```

Guardar Borrar

En esta pantalla, al pulsar sobre el NIU de una respuesta, navega al detalle de la misma y permite votarla. En esta pantalla se muestran las notas que ha recibido la respuesta. En la tabla superior se muestra el NIU del usuario que responde, fecha de la respuesta, nota dada por el profesor, nota dada por el alumno que realizó la pregunta y nota media dada por los demás alumnos. En la parte inferior se puede votar seleccionando un valor del 1 al 5, donde 1 es la nota mínima y 5 la máxima, y pulsando el botón “Votar”.

CATEGORIAS

DETALLE DE LA RESPUESTA

NIU	FECHA	Nota Profesor	Nota Alumno	Nota Resto
100033333	2013-12-23 17:02:24.0	5	4	3.00

TEXTO DE LA RESPUESTA

```
<cola> ::= <frente> + <final> + {<nodo>} + << tamaño>> <frente> ::= <enlace> <final> ::= <enlace> <enlace> ::=
<< índice>> // donde índice=0.. tamaño-1 <nodo> ::= <información> + <enlace> <información> ::= << dato>>
{<< dato>>}
```


VOTACION : Calificación de la Respuesta donde 1 es la mínima y 5 la máxima

1 2 3 4 5

Votar

Perfil Profesor

La pantalla principal del perfil profesor muestra una tabla con las 5 preguntas que se han respondido más recientemente. Se puede acceder al detalle de cada pregunta pulsando sobre el título de la misma.



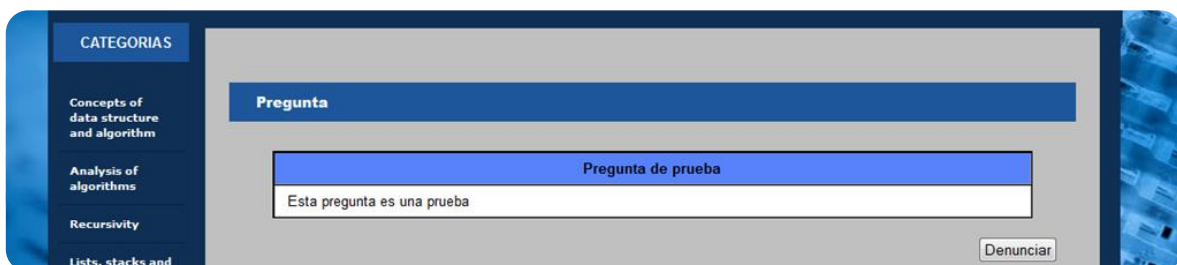
The screenshot shows the EDA (Estructura de Datos y Algoritmos) teacher profile page. The page has a blue header with the EDA logo and the text "EDA Estructura de Datos y Algoritmos". A "Salir" button is in the top right. The left sidebar contains a "CATEGORIAS" section with links to "Concepts of data structure and algorithm", "Analysis of algorithms", "Recursivity", "Lists, stacks and queues", "Trees", and "Java". Below this is an "OPCIONES" section with buttons for "Lista Alumnos" and "Informe Notas". The main content area is titled "LISTADO DE RESPUESTAS RECIENTES" and contains a table with the following data:

TITULO DE LA PREGUNTA	CATEGORIA	ULTIMA RESPUESTA
Implementacion Java	Java	2013-12-23 17:49:29.0
Ejercicio 8 (Hoja 1)	Recursivity	2013-12-23 17:41:16.0
Ejercicio Array	Recursivity	2013-12-23 17:34:01.0
Pregunta de prueba	Concepts of DS and algorithms	2013-12-23 17:19:48.0
Descripcion EBNF	Lists, stacks and queues	2013-12-23 17:02:24.0

Como puede apreciarse en la imagen, la pantalla está compuesta por:

- Una serie de categorías que se corresponden con los temas de la asignatura.

La única diferencia con respecto a las pantallas de Preguntas y Respuestas detalladas en el perfil alumno, es que el profesor tiene la opción de denunciar las preguntas y respuestas que considere que deben borrarse a través del botón "Denunciar".



The screenshot shows the EDA (Estructura de Datos y Algoritmos) teacher profile page with the details of a question. The page has a blue header with the EDA logo and the text "EDA Estructura de Datos y Algoritmos". A "Salir" button is in the top right. The left sidebar contains a "CATEGORIAS" section with links to "Concepts of data structure and algorithm", "Analysis of algorithms", "Recursivity", "Lists, stacks and queues", "Trees", and "Java". Below this is an "OPCIONES" section with buttons for "Lista Alumnos" and "Informe Notas". The main content area is titled "Pregunta" and contains a blue box with the text "Pregunta de prueba". Below this is a text input field with the text "Esta pregunta es una prueba". A "Denunciar" button is in the bottom right corner.



➤ Las opciones propias del perfil:

- Lista de alumnos, donde puede validar o solicitar el borrado de alumnos seleccionando los alumnos y pulsando el botón que corresponda.



- Informe Notas, donde permite calcular la nota final de un alumno en función de las votaciones recibidas a las respuestas dadas por dicho alumno. La nota final de un alumno se calculará en función de la combinación de tres criterios distintos:

- Media de las votaciones que el alumno recibió por parte del profesor.
- Media de las votaciones que el alumno recibió por parte del alumno que realizó la pregunta.
- Media de las votaciones que el alumno recibió por parte del resto de alumnos.

El profesor podrá elegir qué peso da a cada uno de estos criterios a la hora de calcular la nota final, teniendo en cuenta siempre que la suma de los valores de los combos no puede superar el 100%. Por ejemplo, el profesor podría decidir considerar únicamente las votaciones hechas por él (asignando entonces un 100%), o considerar también las votaciones realizadas por el alumno que realizó la pregunta y por las votaciones recibidas por el resto de alumnos con igual peso (asignando entonces a cada criterio un 33%). Al pulsar el botón “Calcular Notas” se actualizará el valor de la columna Final en la tabla. El profesor podrá simular distintas combinaciones y repetir el cálculo de la nota final tantas veces como desee, antes de enviar las notas definitivas.

El botón “Enviar Notas Finales” permite enviar la nota final al correo de los alumnos.

CATEGORIAS

Concepts of data structure and algorithm

Analysis of algorithms

Recursivity

Lists, stacks and queues

Trees

Java

ACCESO

Lista Alumnos

Informe Notas

LISTADO DE NOTAS

En la siguiente tabla se muestran las votaciones recibidas por alumno.

Profesor: Media de votaciones recibidas por el profesor.
Alumno: Media de votaciones recibidas por el alumno que realiza la pregunta.
Resto: Media de votaciones recibidas por el resto de alumnos.
Final: Media final calculada a partir de las anteriores.

NIU	Nombre y Apellidos	NOTAS			
		Profesor	Alumno	Resto	Final
100033333	Pedro Calvo Martinez	3,50	3,50	3,67	0.0
100055555	Laura Diaz Peña	3,00	3,00	5,00	0.0

Seleccione las opciones a partir de las que desea calcular la nota final:

☐ Profesor 25 %
 ☐ Alumno 25 %
 ☐ Resto 25 %

Calcular Notas

Enviar Notas Finales

Perfil Administrador

Para acceder como administrador existe un usuario cuyos datos son admin/admin (NIU/contraseña).

La pantalla principal del perfil administrador es la que se muestra a continuación y corresponde con la funcionalidad de dar de baja a alumnos. En la tabla aparecerán los alumnos que el profesor ha solicitado que se borren. Al pulsar el botón “Borrar Alumnos” se eliminarán todos los que aparecen en la tabla.

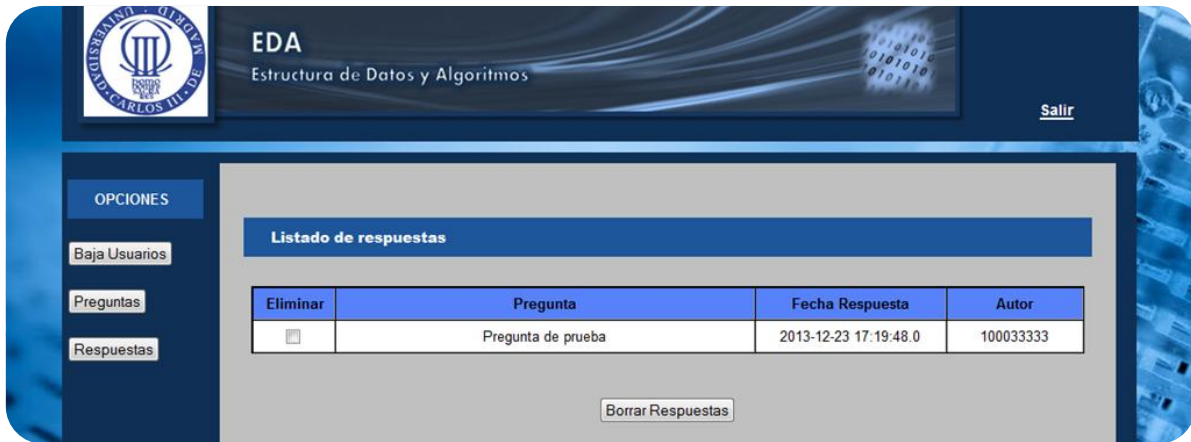


Como puede apreciarse en la imagen, la pantalla está compuesta por las opciones propias del perfil:

- Borrar preguntas, donde se muestran las preguntas que han sido denunciadas por el profesor. Para eliminarlas hay que seleccionar en la columna Eliminar los checkbox y pulsar el botón “Borrar Preguntas”.



- Borrar respuestas, donde se muestran las respuestas que han sido denunciadas por el profesor. Para eliminarlas hay que seleccionar en la columna Eliminar los checkbox y pulsar el botón “Borrar Respuestas”.



- Borrar alumnos, donde se pueden eliminar los alumnos que el profesor haya solicitado a través de su perfil. Coincide con la pantalla inicial.

Para cerrar la sesión de un usuario se debe pulsar el botón *Salir* que se encuentra en la parte superior derecha de todas las pantallas de cada perfil.